NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

NETWORK-BASED CONTROL, MONITORING AND CALIBRATION OF SHIPBOARD SENSORS

by

Eusébio Pedro da Silva

September 2003

Thesis Advisor:	Xiaoping Yun
Second Reader:	Fotis Papoulias

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Network-Based Control, Monitoring and Calibration of Shipboard Sensors			**5. FUNDING NUMBERS**
6. AUTHOR(S) Silva, Eusebio P.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The objective of this thesis is to develop a new calibration system for analog and smart digital pressure sensors, operable by only one person, and capable of calibrating local and remote sensors connected via RS232 cables, Bluetooth or an 802.11b wireless LAN. It is proposed that the operator uses a portable calibration standard and a tablet PC to conduct the sensor calibration. In order to handle local sensors directly connected to the tablet PC and remote sensors connected to the tablet PC via a network capable application processor (NCAP), a dual module application is proposed and developed using LabVIEW. The application has a Master Module and a Slave Module. Both modules are able to connect to multiple digital sensors at the same time. The Master Module was designed to run on the operator's tablet PC offering an easy-to-use graphical user interface (GUI) that allows the monitoring or calibration of any connected sensors. The Slave Module was designed to run on any networked PC, including the operator's tablet and an NCAP. A dedicated Virtual Instrument (VI) was designed for an iterative calibration process based on a least squares fitting method. This VI automatically computes the calibration constants that minimize the measurements errors, and writes the calibration constants to the sensor's RAM or EEPROM. A prototype shipboard sensor test bed was constructed in the laboratory, which consists of a Honeywell PPT digital pressure sensor, an Omega analog pressure sensor, and other 802.11b and Bluetooth wireless LAN components. The newly developed calibration system was successfully demonstrated.

14. SUBJECT TERMS Network-Based Calibration, Wireless LAN, Bluetooth, Datasocket, Labview, Smart Sensors, Pressure Sensors			**15. NUMBER OF PAGES** 83
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

NETWORK-BASED CONTROL, MONITORING AND CALIBRATION OF SHIPBOARD SENSORS

Eusebio Pedro da Silva
Lieutenant Commander, Brazilian Navy
B.S., Escola Naval (Brazil), December 1990
B.S.E.E., Universidade de São Paulo (Brazil), December 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Eusébio Pedro da Silva

Approved by: Xiaoping Yun
Thesis Advisor

Fotis A. Papoulias
Second Reader

John P. Powers
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The objective of this thesis is to develop a new calibration system for analog and smart digital pressure sensors, operable by only one person, and capable of calibrating local and remote sensors connected via RS232 cables, Bluetooth or an 802.11b wireless LAN.

It is proposed that the operator uses a portable calibration standard and a tablet PC to conduct the sensor calibration. In order to handle local sensors directly connected to the tablet PC and remote sensors connected to the tablet PC via a network capable application processor (NCAP), a dual module application is proposed and developed using LabVIEW. The application has a Master Module and a Slave Module. Both modules are able to connect to multiple digital sensors at the same time. The Master Module was designed to run on the operator's tablet PC offering an easy-to-use graphical user interface (GUI) that allows the monitoring or calibration of any connected sensors. The Slave Module was designed to run on any networked PC, including the operator's tablet and an NCAP. A dedicated Virtual Instrument (VI) was designed for an iterative calibration process based on a least squares fitting method. This VI automatically computes the calibration constants that minimize the measurements errors, and writes the calibration constants to the sensor's RAM or EEPROM.

A prototype shipboard sensor test bed was constructed in the laboratory, which consists of a Honeywell PPT digital pressure sensor, an Omega analog pressure sensor, and other 802.11b and Bluetooth wireless LAN components. The newly developed calibration system was successfully demonstrated.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. MOTIVATION....1
B. OBJECTIVE OF THESIS RESEARCH....2
C. DESCRIPTION OF CHAPTERS IN THESIS....2

II. HARDWARE DESCRIPTION....3
A. THE WISER 2400IP ADAPTER [AFTER REF. 4]....3
1. Main Specifications of the WiSER 2400IP Adapter [Ref. 4]....4
2. Use in Thesis....5
B. HONEYWELL PRECISION PRESSURE TRANSDUCER (PPT)....5
1. Main Characteristics [After Ref. 3]....6
2. Use in Thesis....6
C. OTHER COMPONENTS....7
1. The 3E-550I Industrial Wireless Input/Output Node (W-LION)....8
2. 3ETI Industrial Gateway....8
3. 3eTI Bluetooth Module....8
4. Laptop or Tablet PC....8
5. The Portable Pressure Calibrator (PPC), the Digital Sensor SiPC6 and the Pump....9
D. SUMMARY....9

III. SOFTWARE ARCHITECTURE....11
A. PROBLEM STATEMENT....11
B. PROPOSED APPROACH....12
C. USE OF LABVIEW....14
D. USE OF DATASOCKET AS THE COMMUNICATION CHANNEL BETWEEN MODULES....14
E. LOCAL SENSORS VS. REMOTE SENSORS....16
F. SUMMARY....16

IV. THE MASTER MODULE....17
A. INTRODUCTION....17
B. MULTI-THREAD FEATURES....17
C. THE SENSOR CONNECTION PROCESS....19
D. THE CALIBRATION PROCESS....22
1. Background....22
2. The Honeywell PPT Calibration Process....26
a. Canceling the Current Compensation Block Prior to the Application of the LSF Method....27
b. Applying the LSF to a Set of Points (y_c, y_s) to Obtain a_2 and b_2 ($y_s = a_2 y_c + b_2$) in Order to Compute a and b....27

c. Applying the LSF to the Readings (y_s, y_c) to Obtain a_3 and b_3 ($y_c = a_3 y_s + b_3$) in Order to Compute a and b 28
3. A Digital Sensor Calibration Example 31
a. Solution Using the First Approach 32
b. Solution Using the Second Approach 33
c. Solution Using the Third Approach 34
E. MAIN SUBVIS OVERVIEW 36
F. COMMUNICATION THROUGH THE COMMAND-REPLY CHANNEL 39
G. SUMMARY 41
V. THE SLAVE MODULE 43
A. INTRODUCTION 43
B. MULTI-THREAD FEATURES 43
C. MAIN SUBVIS OVERVIEW 44
D. SUMMARY 48
VI. THE NEW CALIBRATION SYSTEM FOR ANALOG SENSORS 49
A. INTRODUCTION 49
B. THE MAIN VI AND THE USER INTERFACE 49
C. VIs RUNNING ON THE NCAP 53
D. THE CALIBRATION PROCESS 55
E. SUMMARY 57
VII. CONCLUSIONS 59
A. SUMMARY 59
B. FUTURE WORK 60
LIST OF REFERENCES 61
INITIAL DISTRIBUTION LIST 63

LIST OF FIGURES

Figure 1 WISER 2400IP Adapter [From Ref. 4].....3
Figure 2 WiSER 2400IP Adapter's Dataflow. 4
Figure 3 The Honeywell PPT. [From Ref. 3]. 7
Figure 4 Basic Wireless Local Area Network (W-LAN) Topology [From Ref. 6]. 12
Figure 5 Master and Slave Modules in (Left) Single or (Right) Dual Machine Configuration. 13
Figure 6 Datasocket Server and the Publisher/Subscriber Relationship. 15
Figure 7 Application Graphical User Interface (Master Module). 18
Figure 8 Threads to Gather and Display Data from Remote and Local Sensors. 18
Figure 9 Example of a Sensor Cluster. 20
Figure 10 Modal Dialog to Append a Sensor to Sensor Set. 20
Figure 11 Diagram of RequestNewSensorInfo.vi. 21
Figure 12 Diagram of ConnectSensor.vi. 22
Figure 13 Transfer Curve After Linear Fitting in Example 4.1.1. 25
Figure 14 Introduction to Digital Sensor Calibration. 26
Figure 15 Custom Slope and Offset Options [From Ref. 3]. 30
Figure 16 Calibration.vi During Honeywell PPT Calibration Example 4.3.1 31
Figure 17 Sensor and Calibration Standard before Calibration. 35
Figure 18 Sensor and Calibration Standard after Calibration when the Actual Pressure is 5.14 psi. 35
Figure 19 Sensor and Calibration Standard after Calibration when the Actual Pressure is 39.57 psi. 36
Figure 20 Master Module sub-VIs Hierarchy. 36
Figure 21 Diagram of CommandReply.vi. 40
Figure 22 Threads Running in the Slave Module after Initialization. 44
Figure 23 Slave Module Hierarchy Tree. 44
Figure 24 Slave Module Main sub-VI (SlaveCore.vi). 46
Figure 25 Diagram of SelectFields.vi. 47
Figure 26 Diagram of IDtoIndex.vi. 47
Figure 27 Block Diagram of the Analog.vi. 50
Figure 28 Front Panel of the Analog.vi. 51
Figure 29 Log File Generated or Updated at the End of Calibration Process. 52
Figure 30 The ICAS.txt File. 53
Figure 31 Diagram of perchs_daq1_local.vi [From Ref. 5]. 54
Figure 32 Diagram of HandleNCAPConsts.vi. 54
Figure 33 User Interface for the Calibration Process. 56
Figure 34 OmegaCalib.vi Block Diagram. 56
Figure 35 Front Panel of Analog.vi After the Calibration Process. 57

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1	Temperature vs. Output Voltage for Transducer T3.	24
Table 2	Summation Table for Example 4.1.1.	25
Table 3	Data for the Honeywell PPT Calibration Example 4.3.1.	31
Table 4	Conversion Table for Example 4.3.1.a	32
Table 5	Summation Table for Example 4.3.1.a.	32
Table 6	Summation Table for Example 4.3.1.b.	33
Table 7	Summation Table for Example 4.3.1.c.	34
Table 8	Summary of the Key VI's Designed for the Main Module.	38
Table 9	List of Slave Module's Commands.	41
Table 10	Summary of subVIs Designed for the Slave Module.	45

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS

CBM	Condition Based Maintenance
DS	DataSocket
GUI	Graphical User Interface
HM&E	Hull, Mechanical and Electrical
HMI	Human Machine Interface
ICAS	Integrated Condition Assessment System
LBTF	Land-Based Testing Facility
MCS	Machinery Control Systems
MM	Master Module
NCAP	Network Capable Application Processor
NPS	Naval Postgraduate School
PPC	Portable Pressure Calibrator
PPT	Precise Pressure Transducer
RCM	Reliability Centered Maintenance
SM	Slave Module
SubVI	Subsidiary Virtual Instrument
VI	Virtual Instrument
W-LION	Wireless Input/Output Node

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my beloved wife Adriana for her continuous support during the moments we have been together and for understanding the times we had to be apart. Her dedication and love has given me the strength I needed to face the most difficult situations in life. The birth of our daughter Laura during my time at NPS was one of the most remarkable experiences of my life and an incentive to continue working and improve our quality of life.

To Professor Xiaoping Yun I would like to express my deep gratitude for his guidance and patience during the development of this thesis. His assistance helped me to overcome all the difficulties encountered along the way. Thanks for the knowledge you have passed on to me since our first lectures and for your invaluable friendship.

I also would like to thank Professor Fotis Papoulias for his support and brilliant ideas.

Special thanks also goes to Brazilian Navy officers CDR Araujo, CDR Joaquim and LCDR Alexandre Mendonça for their support.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The objective of this thesis is to develop a new calibration system for analog and smart digital sensors, which does not requires more than one person to fully operate and is capable of calibrating local and remote sensors connected via RS232 cables, Bluetooth or 802.11b wireless LAN.

In order to handle remote sensors connected to shipboard Network Capable Application Processors (NCAPs) as well as local sensors directly connected to the operator's computer, a dual module application is proposed and developed using LabVIEW 6.0. The application has a Master Module and a Slave Module that can run in the same or different machines exchanging information used by the monitoring and calibration processes. As far as communication and interface are concerned, this application equally treats sensors permanently installed on ships for measurement purposes and calibration standards used to calibrate other sensors. Calibration standards are usually connected to the application as local sensors.

This dual module application was designed for the typical situation where an operator goes to the calibration site with a laptop or a tablet PC and locates the sensors scheduled for calibration already connected to an NCAP. If the NCAP is running the Slave Module, the operator can wirelessly control all the sensors already connected and additionally connect other sensors to the laptop or tablet PC, e.g., a calibration standard. Hence, the shipboard sensors do not need to be disconnected from the NCAP to be calibrated.

Both Master Module and Slave Module are able to connect and handle an unlimited number of digital sensors. The Master Module was designed to run on the operator's laptop or tablet PC offering an easy-to-use GUI that allows the monitoring or calibration of any connected sensors. The Slave Module was designed to run on any machine connected to the WLAN, including the operator's laptop. A dedicated VI was designed for an iterative calibration process based on a least squares fitting method and is able to automatically compute the calibration constants that minimize the measurements errors through the sensor full scale. The VI is also able to write the calibration constants to the sensor's RAM or EEPROM.

A prototype shipboard sensor test bed was constructed in the laboratory, which consists of a Honeywell PPT digital pressure sensor, an Omega analog pressure sensor, and other 802.11b and Bluetooth wireless LAN components. The newly developed calibration system was successfully demonstrated to NAVSEA-Corona and NAVSEA-Philadelphia representatives. A lighter version was developed to streamline the current shipboard calibration procedures for analog sensors, reducing the time and personnel required for periodical calibration tasks. The application is scheduled to be tested and evaluated in the Land-Based Testing Facility (LBTF) at NAVSEA-Philadelphia.

I. INTRODUCTION

A. MOTIVATION

Naval ships deploy many sensors to monitor shipboard system conditions and environmental conditions. In order to reduce the requirement for shipboard manning, many more sensors are required for shipboard monitoring. Current DDG ships have approximately 3,742 hull, mechanical, and electrical (HM&E) sensors [Ref. 1]. The number of sensors required for a next generation destroyer, DD21, is expected to be in the range of 200,000 [Ref. 2].

The reliable operation and readiness of ships depend on the accuracy of sensor data. To ensure accurate readings from sensors over long periods of time, most sensors need to be calibrated on a regular basis. The current calibration process is mostly manual, time consuming, and labor intensive [Ref. 1]. To meet the challenge of reducing crew size and significantly deploying more sensors in the future, there is a clear need for developing new, automated calibration processes and standards.

While mechanical gages and analog sensors prevail on today's ships, advanced sensors such as IEEE 1451 smart sensors and wireless sensors are expected to be installed on future ships. For the most part, calibration processes for smart sensors have not been established. Conventional analog sensors simply provide a voltage or current output proportional to pressure, temperature, or other physical parameters to be measured. Calibration constants are stored elsewhere outside of sensors, e.g., at the Integrated Condition Assessment System (ICAS). In addition to providing measurement readings, smart sensors have other features, including the capability of storing calibration constants within sensors themselves. Honeywell PPT (Precision Pressure Transducer) is an example of smart sensors [Ref. 3]. It provides pressure measurements in both analog and digital forms and has several programmable features such as selectable pressure units, programmable integration time, automatic read rate adjusting, network traffic reduction, addressable network, adjustable calibration constants, retaining new configurations in RAM or EEPROM, and others.

B. OBJECTIVE OF THESIS RESEARCH

The objective of this thesis was to develop a closed-loop calibration system for smart digital sensors. The calibration system provides flexibility for calibrating sensors with different types of connection configurations, including local sensors, remote sensors, sensors connected with RS232 cables, sensors with Bluetooth wireless connections, and sensors connected through network ports. The calibration system is backward compatible to calibrate analog sensors as well.

C. DESCRIPTION OF CHAPTERS IN THESIS

Chapter I defines the objective of this thesis by giving a brief explanation about the current use of sensors in shipboard systems and the new tendency regarding the use of digital sensors. Chapter II details the hardware used in this thesis. Chapter III describes the two module approach proposed to solve the problem and explains the software architecture. Chapter IV explains the Master Module's design and its major features. Chapter V explains the Slave Module's design and its major features. Chapter VI presents a simplified calibration system focusing on the monitoring and calibration of shipboard analog sensors. Chapter VII presents the conclusions and explains the accomplishments of the thesis.

II. HARDWARE DESCRIPTION

This chapter presents hardware components used in this thesis. The equipment used includes the WiSER 2400IP adapter, the Honeywell Precision Pressure Transducer (PPT), the 3e-550I Industrial Wireless Input/Output Node (W-LION), the 3eTI Industrial Gateway, the 3eTI Bluetooth Module, the laptop or tablet PC, the Portable Pressure Calibrator (PPC), the digital sensor SiPC6 and the pump.

A. THE WISER 2400IP ADAPTER [AFTER REF. 4]

The WiSER 2400IP is an 802.11b compliant, or WiFi, radio with an RS232 serial interface. See Figure 1.

Figure 1 WISER 2400IP Adapter [From Ref. 4].

Figure 2 illustrates the WiSER 2400IP dataflow when transmitting and receiving data to and from the wireless network.

When transmitting to the network, the WiSER 2400IP radio takes serial data from the equipment or computing device connected via its RS232 port, converts the serial data into TCP/UDP data packets, and transmits these packets with the RF modulation that is compliant with the specifications of the 802.11b physical layer.

When receiving data from the wireless network, the radio demodulates the RF signal, removes the Ethernet headers, unpacks the packet and delivers the data byte-by-byte to the destination equipment/device through the RS232 serial port.

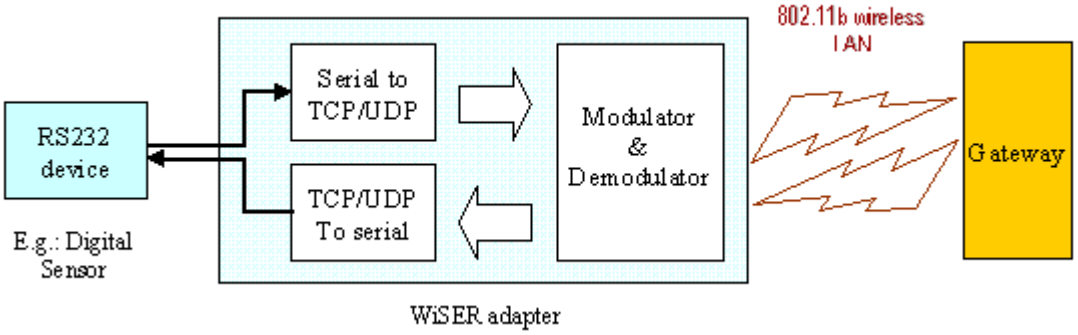

Figure 2 WiSER 2400IP Adapter's Dataflow.

Each WiSER 2400IP radio acts as a "Station" and operates in either infrastructure or ad-hoc mode in accordance with the 802.11 standards. As such, this radio enables RS232-interfaced devices to participate in a wireless Ethernet network. In this capacity, the radio, in addition to eliminating the RS232 cables, functions as a media converter for RS232-interfaced equipment and IP-based computing-devices.

The radio is fully self-contained in performing the conversion between serial data and wireless Ethernet packets. That is, no device driver needs to be installed on the hosting equipment or computing device to which the radio is connected. The True Plug and Play feature, therefore, is achieved with any equipment or computing devices with a RS232 port. This also means the radio can be used on equipment and/or computing-devices with any operating system. This is particularly useful for instruments/equipment where the use of a RS232 interface is utilized.

1. Main Specifications of the WiSER 2400IP Adapter [Ref. 4]

- Standard: 802.11b
- Host Interface: RS232
- Frequency: 2.4 GHz – 2.495 GHz
- Link Distance: ~1200 ft in open space
- Data Encryption: Support the standard 64-bit and 128-bit WEP

- Network Security: MAC-address-based access control

2. Use in Thesis

In this thesis, the WiSER 2400IP will be used to connect smart digital sensors to the wireless Ethernet network in accordance with the 802.11b standards. The application shall allow the establishment of communication channels using TCP or UDP protocols between each WiSER 2400IP and the operator's laptop/tablet PC. Additionally, when the application is running in a dual machine configuration, the WiSER 2400IP can also be connected directly to the NCAP (see Chapter III).

The WiSER 2400IP adapter makes it possible to eliminate the RS232 cables, to surpass the limitation of the number of serial ports per machine, and to pass over the NCAP itself because the smart sensors' information can be directly delivered to the client application (ICAS).

B. HONEYWELL PRECISION PRESSURE TRANSDUCER (PPT)

The Honeywell precision pressure transducer (PPT) is a "smart sensor" that combines silicon sensor technology with microprocessor-based signal conditioning.

The heart of this digital sensor is a silicon piezoresistive transducer which contains both pressure and temperature-sensitive elements. Digital signals representing temperature and pressure are processed by a microprocessor to produce fully temperature compensated and calibrated pressure readings over the entire –40 to 85 °C (–40 to 185 °F) temperature range.

The user is not allowed to modify the factory calibration major settings. However, small adjustments (±0.6%FS in 0.005% increments) can be made to the pressure transfer curve by modifying the PPT's full scale slope and offset.

The factory calibration, together with the user's adjustments allows, in the majority of the cases, the perfect calibration of the sensor. This feature is explored to implement the closed-loop calibration system as described in Chapter IV.

To allow the user to retrieve and store the full scale slope and offset values, the Honeywell PPT has specific commands. Additionally, it offers many others commands

used to select the type of measurement (pressure, temperature in °F, temperature in °C), units of pressure, output mode, read out resolution, sample rates, baud rates and other choices.

There are also commands to set the PPT either to a temporary configuration, until the PPT is powered down, or to a permanent configuration if stored in the internal EEPROM. In this case, the stored settings are automatically loaded each time power is applied.

1. Main Characteristics [After Ref. 3]

- Digital Accuracy (from –40° to 185° F): ±0.05 % FS Typical
- Analog Accuracy (from –40° to 185° F): ±0.06 % FS Typical
- Temperature Accuracy (from –40° to 185° F): ±1°C (at sensing element)
- Operating Temperature Range: –40° to 185°F
- Sample rate: 8.33 ms to 51.2 min
- Digital Resolution: Up to 0.0011 % FS
- Analog Resolution: 1.22 mV Steps (12 bits)
- Response Delay: 1 ms, maximum 17 ms
- Long term stability: 0.02 % FS max per yr.
- Digital Output: RS-232
- Analog Output: 0 to 5 V
- Baud rate: 1200, 2400, 4800, 9600, 14400, 19200, 28800.

2. Use in Thesis

This thesis uses the Honeywell PPT digital sensor as the main target sensor for the closed-loop calibration process. Although other types of sensors can be used in the digital calibration process, the concepts presented here can be easily extended to other digital sensors.

The Honeywell PPT is used in all different manners provided by the application, connected via an 802.11b wireless LAN attached to a WiSER 2400IP adapter, Bluetooth or RS232 cables.

Even if the application is in dual or single machine configuration, the sensor can be connected to either a Master or Slave Module to be calibrated, monitored or used as a calibration standard.

Hence, the Honeywell PPT is used not only to prove that the application is capable of performing the calibration of digital sensors, but also to explore the application's flexibility and monitoring capabilities.

During the calibration process, if it is used as the target sensor, another sensor is required as the calibration standard. On the other hand, if it is chosen as the calibration standard, the user can benefit from its high accuracy to calibrate any other locally or remotely connected sensors.

Figure 3 The Honeywell PPT. [From Ref. 3].

C. OTHER COMPONENTS

In addition to the WiSER adapter and the Honeywell PPT, this thesis uses a number of other hardware components. These components were described in detail in another thesis [Ref. 5]. A brief description of these components is provided below.

1. The 3E-550I Industrial Wireless Input/Output Node (W-LION)

The (W-LION) is a Network Capable Application Processor (NCAP) with a compacted and ruggedized case, suitable for working in severe environments. Currently, the W-LION has been used in shipboard systems to run a data acquisition and managing program to acquire data, process the raw data and send the information to a wireless access point which transports it to a client application located elsewhere in the system architecture [Ref. 6].

2. 3ETI Industrial Gateway

The 3e-521N is a wireless dual mode gateway [Ref. 7]. Similar to the W-LION, it also has a rugged water and corrosive proof encasement developed for harsh environments and meets military standards for installation onboard United States Navy ships. It can be configured in either access point mode to bridge wireless users to wired resources or as a gateway to provide additional firewall protection along with multiple broadband media selections. However, in this thesis, it is used in the access point mode together with a 3COM 3CR856-95 router to simulate a ship network.

3. 3eTI Bluetooth Module

The 3e-250 Bluetooth to RS-232 cordless adapter is a small portable serial port to Bluetooth converter. It converts the data flow from a RS-232 serial port connection to Bluetooth protocol and transmits to other Bluetooth adaptors. With this module, the RS232 devices (Honeywell PPT or Crystal PPC) can be wirelessly connected to the laptop or tablet PC as long as they have the Bluetooth adaptor.

4. Laptop or Tablet PC

The laptop/tablet PC provides the user interface to the entire process running the application's Master Module. It also can run the Slave Module when configured in the single machine mode. The operator can use the laptop/tablet PC to calibrate any smart sensor connected to the application by choosing a calibrator from the sensor set.

To be able to control the NCAP when using the dual machine configuration, the laptop/tablet PC also contains a program called vncviewer.exe. This program enables an operator to view and control the desktop of the NCAP from the laptop/tablet PC.

5. The Portable Pressure Calibrator (PPC), the Digital Sensor SiPC6 and the Pump

The portable pressure calibrator (PPC) [Ref. 8], manufactured by Crystal Engineering Corporation, and the digital sensor SiPC6, made by SI Pressure Instruments Ltd. [Ref. 9], are also used in this thesis for monitoring and calibration purposes. As with the Honeywell PPT, these sensors can also be either local or remotely connected via Bluetooth, WiSER 2400IP or RS232 cable.

During the calibration process, a lightweight handheld pump is used to provide the desired pressure through a pneumatic or hydraulic medium. The maximum pressures that the pump can produce are 600 psi for a pneumatic medium or 10,000 psi for a hydraulic medium [Ref. 8].

D. SUMMARY

In this chapter, all the hardware components used in this thesis were presented. The equipment includes the NCAP, Gateway, Bluetooth-to-RS-232 adaptor, laptop/tablet PC, digital sensor SiPC6, Honeywell PPT, WiSER 2400IP, Crystal PPC and the pump. The next chapter presents the software architecture.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SOFTWARE ARCHITECTURE

This chapter discusses the calibration problem and proposes a two modules approach as a flexible solution for a new calibration system. The LabVIEW software is presented as the selected language and some of its features are discussed. The DataSocket interface is also introduced and an initial depiction of the connection process is offered.

A. PROBLEM STATEMENT

Mechanical gauges and analog sensors are widely used in shipboard systems today. In order to preserve reliability, all sensors are required to be periodically calibrated. However, the current calibration process for analog remote display sensors is slow, labor intensive and requires a team of service members to perform the task.

The need for new calibration methods is clear. It is more evident from the fact that new generation ships are expected to have the number of sensors drastically increased, with smart and wireless sensors assuming a major role in the new systems. Furthermore, the challenge to reduce the crew size reinforces the necessity of streamlined calibration methods.

The goal of this thesis was to develop a new closed-loop calibration system for digital smart sensors. The system should be designed to support digital sensors installed in wireless network based shipboard systems which are likely to be used in new generation ships. Figure 4 shows the basic Wireless Local Area Network (W-LAN) topology already tested in the U.S. Navy [Ref. 1]. In this topology, the W-LION (Industrial Wireless Input/Output Node) is used to connect to the sensors and transmit the data via an 802.11b wireless network.

The system must be able to control and calibrate local and remote digital sensors connected via RS232 cables, Bluetooth, or an 802.11b wireless LAN. The sensors are locally connected when they are connected to the operator's laptop or tablet PC and are remotely connected when they are connected to the W-LION (NCAP).

In order to reduce the personnel and time required to conduct the sensor calibration, the system must also allow the operator, without the help of another service member, to control and calibrate any connected sensor from the calibration site.

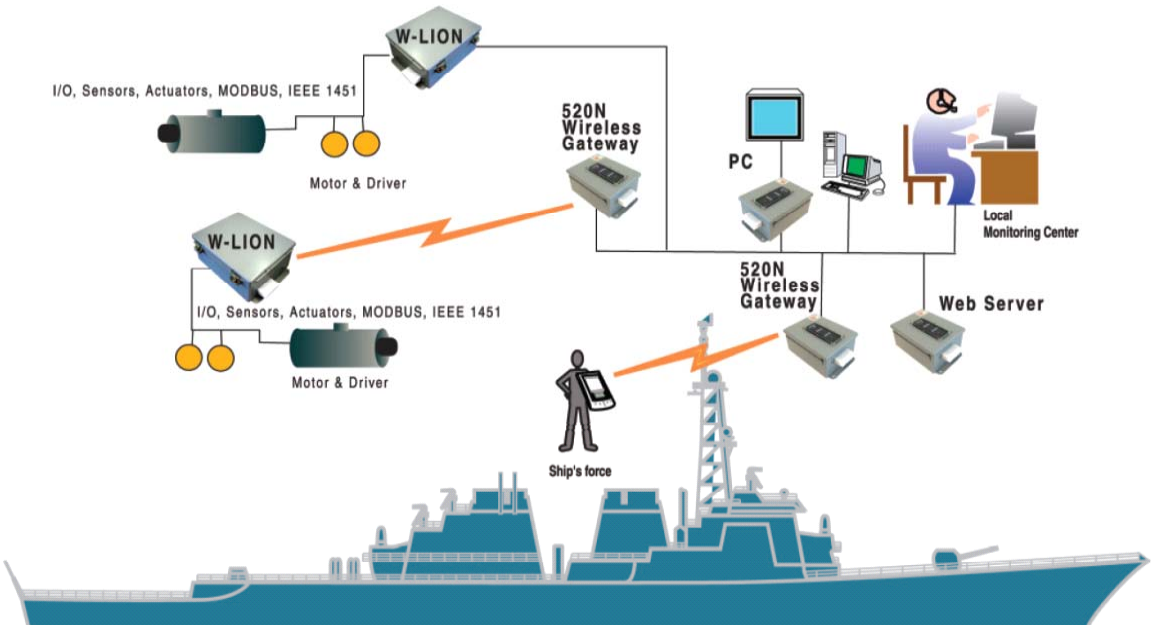

Figure 4 Basic Wireless Local Area Network (W-LAN) Topology [From Ref. 6].

B. PROPOSED APPROACH

The program developed in this thesis is able to perform the monitoring, controlling and calibration tasks of smart sensors connected wirelessly or through a serial port. In order to control sensors connected via NCAP or via the operator's laptop/tablet PC, the application was designed to have two main modules, the Master Module and the Slave Module.

The Slave Module (SM) was designed to collect data from sensors connected wirelessly or through the serial port (RS232). However, this module does not have a user interface and cannot be directly controlled. It relies on the Master Module (MM) to receive the user commands or to display sensor data.

The Master Module plays a more perceptible role because it controls the entire monitoring and calibration process. To perform this role, the Master Module is directly commanded by the user and has a Graphic User Interface (GUI) specifically designed for this purpose. In addition, it also provides a dedicated two-way communication channel to the Slave Module.

This dual-module design offers flexibility to the application, allowing it to be used to control and calibrate shipboard sensors that have their data wirelessly sent to the ship's LAN by a NCAP. With this configuration, each NCAP in the ship can run the Slave Module to gather data from a large set of smart sensors while the Master Module may be running on other machines from which the sensors can be controlled or calibrated. The calibration process, as described in Chapter IV, is faster and more accurate than the calibration process currently used in the Navy.

This dual-module design is suitable for the topology depicted in Figure 4 and the design is not restricted to it. Actually, when the smart sensors are wirelessly connected to the LAN, the NCAP is not needed anymore. However, the application can still be used to control and calibrate the sensors if both modules are executed in the same machine wirelessly connected to the sensor's network.

Figure 5 shows both the single and dual machine configuration, illustrating how several sensors can be wirelessly connected to the Master and Slave modules. It also shows that the user can only interact with the Master Module.

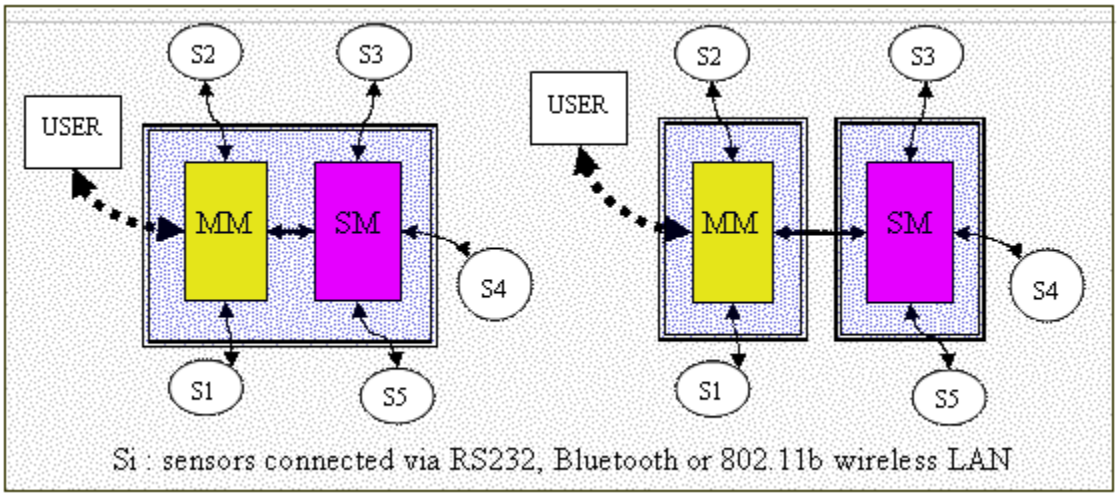

Figure 5 Master and Slave Modules in (Left) Single or (Right) Dual Machine Configuration.

It is important to notice that each digital sensor connected to either Master or Slave Module requires a bi-directional communication channel in order to be able to re-

ceive commands and send replies. Furthermore, during the connection process, the user is required to advise on the type of sensor and connection to be used. Hence, the sensors shown in Figure 5 can represent digital sensors connected via RS232 cables, Bluetooth or an 802.11b wireless LAN attached to a WiSER 2400IP adapter.

To illustrate how the dual-module design can be useful, it is helpful to consider a shipboard system with several smart sensors measuring pressure at different points of the ship and sending the data to a NCAP. If the NCAP is running the SM, the entire calibration process discussed in Chapter IV can be done by one person with a tablet PC or a laptop running the application's MM.

C. USE OF LABVIEW

LabVIEW was selected for the software development in this thesis. It is a multiplatform and dataflow program language with an intensive use of modularity and intuitive diagram representation. The modules are called Virtual Instruments (VIs) and are composed of two windows, the Front Panel and the Block Diagram. The Front Panel is used as the graphical user interface and contains the controls (inputs) and indicators (outputs) used by the user. The Block Diagram window is the place where the code is actually developed. It has terminals associated with all the controls and indicators placed in the Front Panel. During the development of the graphical source code, functions are created and wired to these terminals in order to make operations with data obtained from the controls and to present the results through the indicators. Both windows are linked together in the same VI but are assembled independently, allowing the programmer to focus either on the user interface or on the program functionality.

D. USE OF DATASOCKET AS THE COMMUNICATION CHANNEL BETWEEN MODULES

National Instruments in its measurement suite provides the DataSocket interface. DataSocket is a technology based on the industry standard TCP/IP that pulls together established communication technologies for measurement and automation. It avoids low level programming details by using a self-describing format to transfer data as strings, scalars, booleans or clusters on their original format, eliminating the need for complicated parse code. This high level feature is used to simplify data exchange between the Master and Slave Modules when they are running on the same or on different computers.

To establish the DataSocket communication channel, one application writes the desired data to the Datasocket Server through the DataSocket API. The application that needs the data uses the same API to retrieve the data from the server. Both applications are "clients" from the DataSocket Server point of view. The first one is also called "publisher" and the second one "subscriber".

In this thesis, the Master and Slave modules can act either as publisher or subscriber because two channels were put into effect. The first channel is established just to transfer data gathered from the Slave Module's sensors to the Master Module. The second is a two-way command-reply channel established to allow the Master Module to control the Slave Module's sensors and receive a reply with the desired information or error message. It is important to consider that the Slave Module has only these channels to pass information to the user. Hence, it is not allowed to open a dialog box or request any kind of action directly from the user.

For the purpose of this thesis, the DataSocket communication is restricted to a Local Area Network (LAN). However, communication could be established over the internet by using DataSocket to publish the applications' data as shown in Figure 6. In this case, security becomes an issue and should be treated accordingly by the use of DataSocket access restrictions to administer security and permissions.

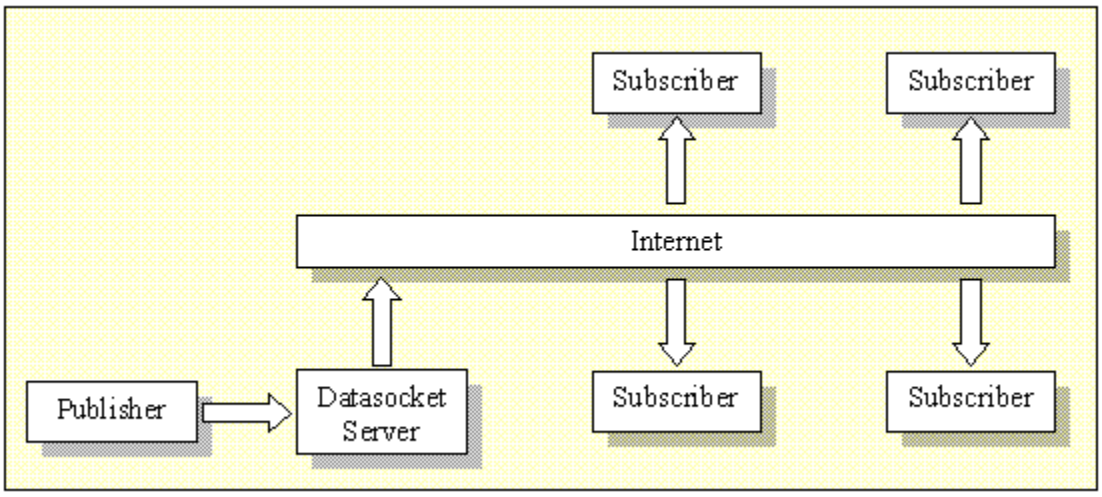

Figure 6 Datasocket Server and the Publisher/Subscriber Relationship.

E. LOCAL SENSORS VS. REMOTE SENSORS

The application developed in this thesis is able to control and monitor a fair number of sensors concurrently. The software design does not impose a limit on the number of sensors.

Typically, the person responsible for the calibration goes to the calibration site with a laptop or a tablet PC and a calibration standard. The digital sensors scheduled for calibration are already remotely connected to the NCAP, and thus are available on the W-LAN. In this typical situation, the operator needs only to connect the calibration standard locally either wired or wirelessly connected to the laptop, after the sensor and calibration standard are physically connected. Hence, the data coming from all shipboard sensors remotely connected to the NCAP can be seen in the operator's laptop and compared with the data coming from the calibration standard locally connected.

When the new calibration system is applied to a shipboard system that uses the topology depicted in Figure 4, only the calibration standard needs to be locally connected to the laptop/tablet PC. However, if the system to be calibrated does not use the NCAP, the application should be in the single machine configuration, with both Master and Slave Module running on the operators laptop/tablet PC. In this situation, the application must be able to connect all the sensors locally via a laptop/tablet PC.

The use of the WiSER 2400IP adapter is one way to have as many locally connected sensors as required. See Chapter II for hardware specifications. The flexibility provided by the dual-module design is further increased by the use of the WiSER 2400IP because each device can be connected to either Master or Slave modules using a TCP or UDP protocol with the WiSER 2400IP acting as a server and the application as a client.

F. SUMMARY

In this chapter the two modules approach is proposed as a flexible solution for a new calibration system. The LabVIEW software is presented as the selected language and some of its features are discussed. An introduction to the connection process is also presented. The next chapter explains how the Master Module works and describes its major features and the calibration process.

IV. THE MASTER MODULE

This chapter presents the major features of the Master Module with emphasis on the multithread characteristics, the sensor connection process, the communication mechanism and the calibration process. The computations required by the calibration process are provided in three different ways with an example showing the details related to each step.

A. INTRODUCTION

The Master Module (MM) is the part of the application directly controlled by the user and has a Graphical User Interface (GUI) especially designed to obtain the user inputs easily and present the results to the user. See Figure 7. Through the MM front panel, the user can add a new sensor to the monitored set, view a graphical history of each monitored sensor, enable or disable the monitoring status of a particular sensor, execute the calibration of a desired sensor against any other sensor chosen as a calibration standard, remove sensors from the set, view the current value of each sensor numerically and graphically, change the type of measurement for the Honeywell PPT, or change the sensor configuration.

B. MULTI-THREAD FEATURES

When the Master Module starts, it goes through an initialization stage and then starts three concurrent threads that will be kept alive until the application is closed by the user. The first thread is responsible for reading and plotting the measurements sent by the Slave Module, remotely if in the dual machine configuration, through the DataSocket channel. The second thread is in charge of the local measurements and it requests the desired measurements to all locally connected sensors and displays the replies graphically and numerically. The third thread implements an event detection mechanism. It is necessary to identify the action requested by the user based on the controls modification. Figure 8 shows the two first threads.

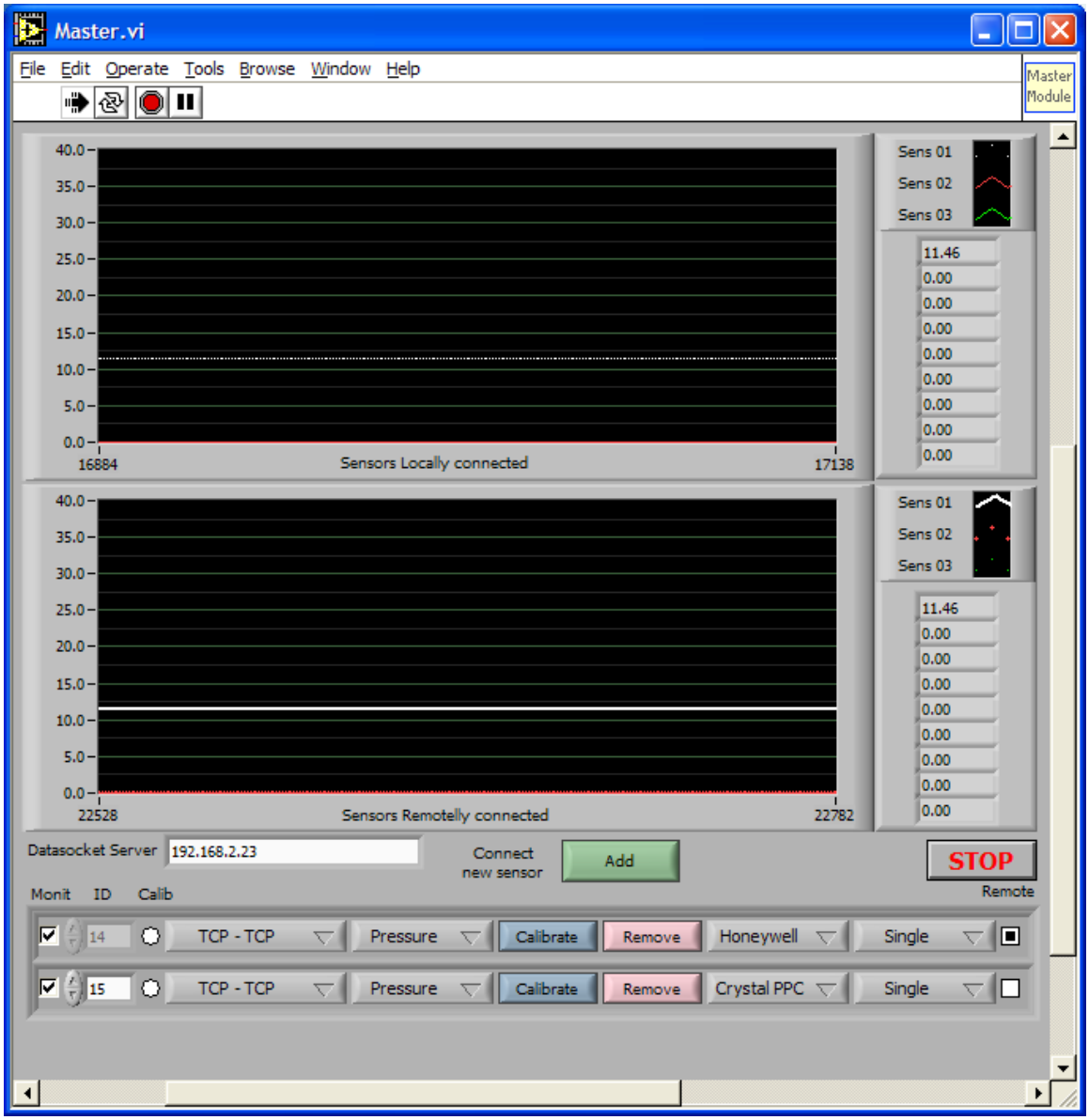

Figure 7 Application Graphical User Interface (Master Module).

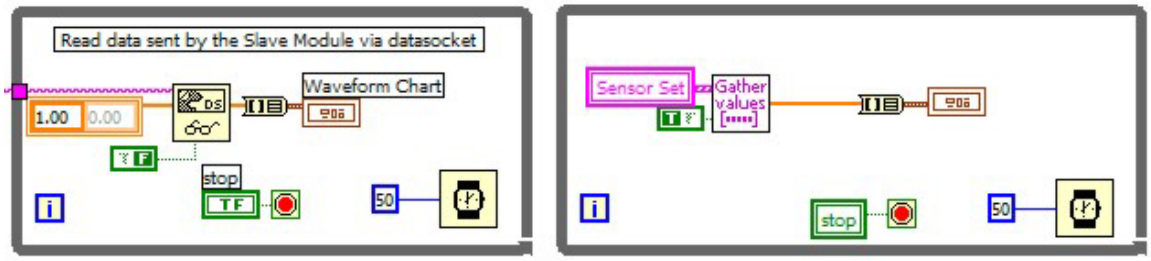

Figure 8 Threads to Gather and Display Data from Remote and Local Sensors.

The third thread is also in charge of the calibration and removal processes, responsible for establishing and maintaining connections with all the local sensors (sensors connected to the Master Module) and for establishing and maintaining the DataSocket connection with the Slave Module which is required to send commands to the remote sensors and receive their replies.

C. THE SENSOR CONNECTION PROCESS

Both Master and Slave Modules use an array of clusters to keep track of the sensors status. This array of clusters, named "Sensor Set", represents a table of remotely and locally connected sensors. Each cluster comprises 19 fields used to store the sensor status as depicted in Figure 9.

When the user presses the "Add" button to append a new sensor to the set, the connection dialog box is displayed, requiring the minimal information for the new sensor. To simplify the connection process, most of the fields are filled with the default values that can only be modified through the GUI when the sensor is already connected. Actually, the user does not need to see all the sensor fields. Some fields are hidden and directly handled by the application.

Through the connection dialog box the user can define if the sensor should be locally or remotely connected or if the connection is wired or wireless. In the case of a wireless connection, the user can use either the Bluetooth or 802.11b standards. If the sensor is attached to a WiSER adapter, the user can choose a TCP or UDP connection. Figure 10 shows the appearance of the modal dialog box when the user requests a TCP connection.

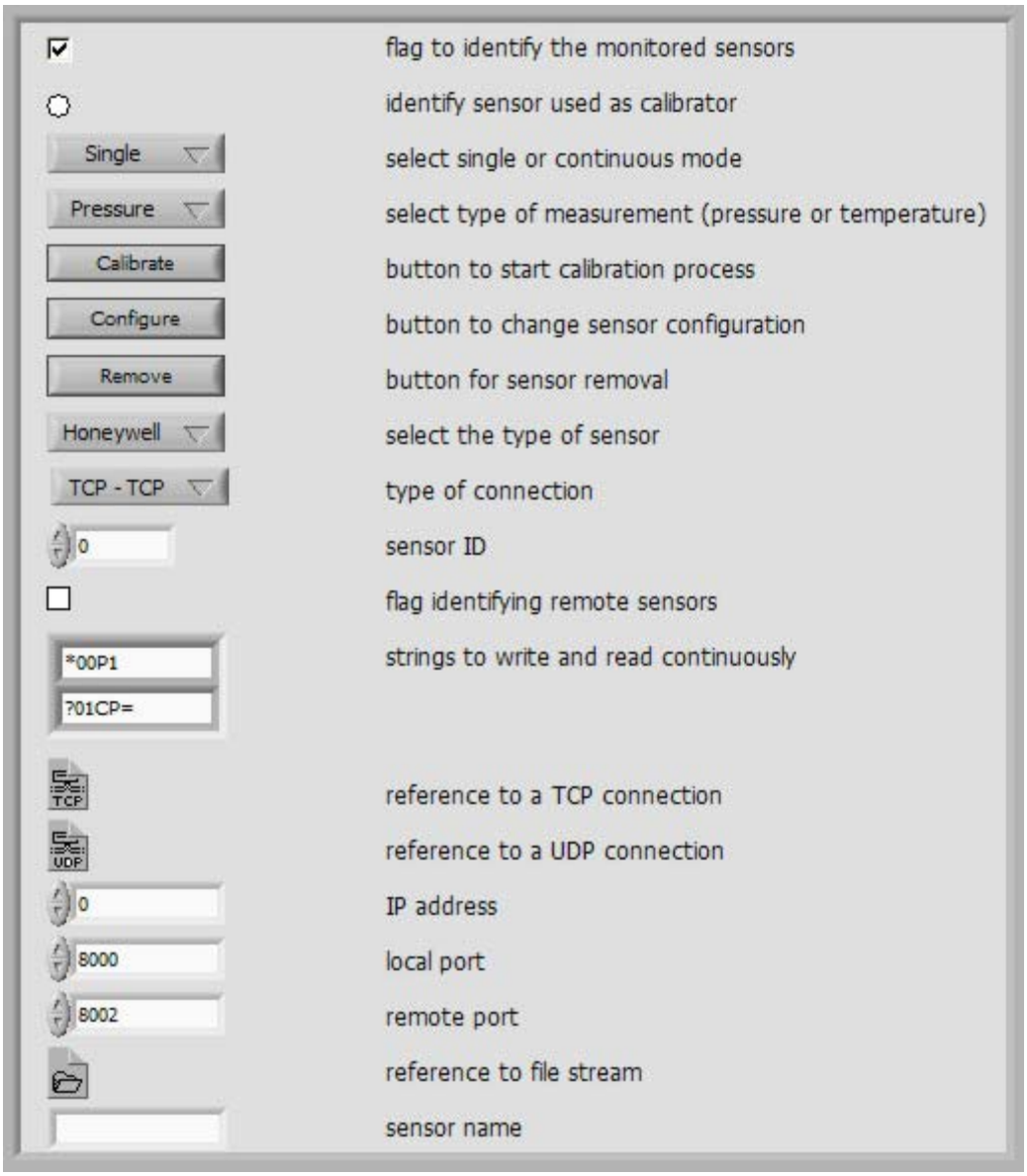

Figure 9 Example of a Sensor Cluster.

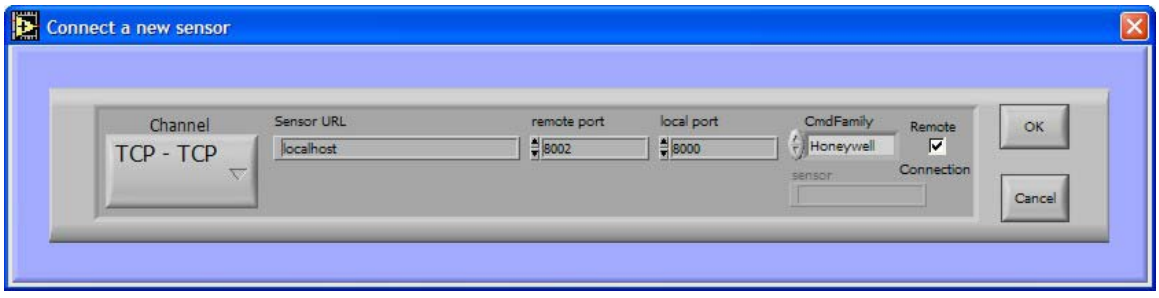

Figure 10 Modal Dialog to Append a Sensor to Sensor Set.

The RequestNewSensorInfo.vi, whose diagram is depicted in Figure 11, is the subVI responsible for the connection dialog box. It is a simple subVI which has a subset of the sensor cluster (the connection fields), containing only a few fields with the minimal information required to establish a sensor connection. When the user chooses the desired channel, the subVI automatically enables the fields related to that specific channel and disables the fields for the information that does not apply to the chosen channel. When all the required information is provided, the user can click the OK button to start the connection process.

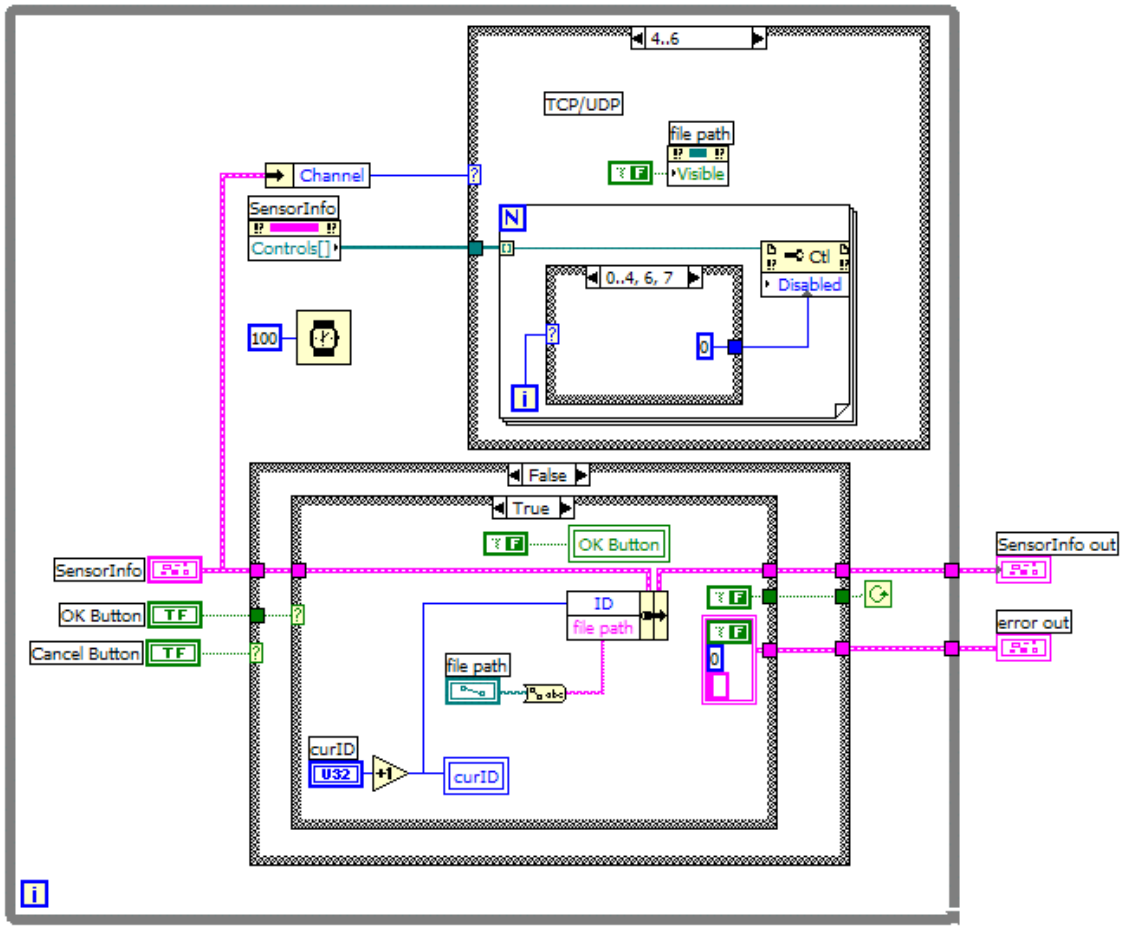

Figure 11 Diagram of RequestNewSensorInfo.vi.

When the user confirms the connection request, the Master Module verifies if the desired connection is remote or local. If the user requests a local connection, the subVI

ConnectSensor.vi, shown in Figure 12, is immediately called. Otherwise, the request is sent to the Slave Module through an "addSensor" command, with the cluster supplied by the connection dialog box.

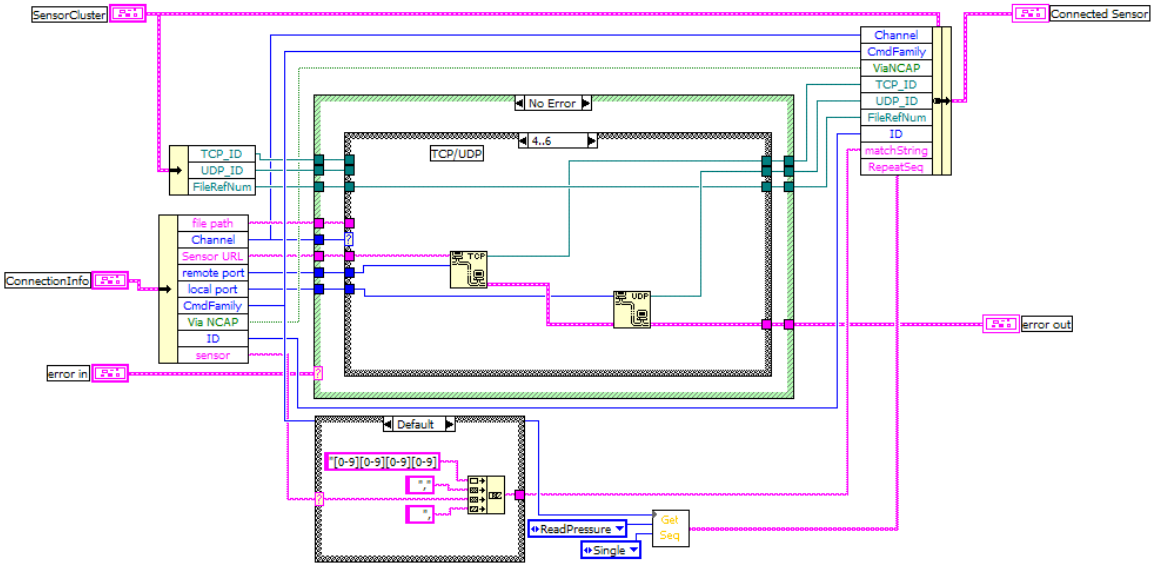

Figure 12 Diagram of ConnectSensor.vi.

The ConnectSensor.vi subVI has the sensor cluster and a cluster with the connection fields as inputs. When a TCP or UDP connection is requested, it takes the sensor URL and the port numbers provided and tries to establish the desired connection. After the connection is established, the reference number generated for the TCP and UDP connection is stored in the sensor cluster which is the main output of the subVI.

D. THE CALIBRATION PROCESS

1. Background

To understand how the smart sensor is calibrated, it is helpful to consider a simple scale conversion case. Consider a linear analog transducer (T1) which produces a voltage of 1 Volt when the temperature is 32°F and 5 Volts when the temperature is 212°F. A transfer curve (TC1) from the transducer output voltage to a Fahrenheit temperature can be found in the following manner:

$$\frac{V-1}{F-32^\circ} = \frac{5-1}{212^\circ-32^\circ} \tag{4.1}$$

which gives

$$F = 45 \times V - 13° \,. \tag{4.2}$$

Now consider that a second transducer (T2) gives a voltage of 0.7 Volt when the temperature is 32°F and 5.2 Volts when the temperature is 212°F. The transfer curve (TC2) for T2 is:

$$\frac{V - 0.7}{F - 32°} = \frac{5.2 - 0.7}{212° - 32°} \tag{4.3}$$

which gives

$$F = 40 \times V + 4° \,. \tag{4.4}$$

Although the transducers T1 and T2 have different output values, the temperature can be precisely determined if the correct transfer curve is applied. Both transfer curves have the form $y = a \cdot x + b$, which is a line with slope a and offset b. TC1 has slope 45 and offset –13° while TC2 has slope 40 and offset 4°.

Given a set of N distinct points (x_i, y_i) related by the expression $y = ax + b$, the slope a and offset b can be determined if $N = 2$. If N>2, the system is over-determined and it may not be possible to find a line which contains all the given points (x_i, y_i). In this case, the line that best fits the points is found. One of the most widely used techniques to solve this problem is the Least Squares Fitting (LSF) [Ref. 10].

The idea behind the Least Squares Fitting is to solve a set of over-determined linear equations such that the summation of the squared deviation is minimized.

The deviation e_i is the difference between the value computed from the fitting curve (i.e., $ax_i + b$) and the actual value y_i. The slope a and offset b that minimize the least squares error can be found by differentiating the expression

$$D(a,b) = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (ax_i + b - y_i)^2 \tag{4.5}$$

which yields

$$\frac{\partial D(a,b)}{\partial a} = \sum_{i=1}^{N} 2(ax_i + b - y_i)x_i = 2\left(a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} x_i\, y_i \right) = 0 \tag{4.6}$$

$$\frac{\partial D(a,b)}{\partial b} = \sum_{i=1}^{N} 2(ax_i + b - y_i) = 2\left(a\sum_{i=1}^{N} x_i + bN - \sum_{i=1}^{N} y_i \right) = 0 \tag{4.7}$$

or in matrix notation

$$\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_i\, y_i \\ \sum_{i=1}^{N} y_i \end{bmatrix} \tag{4.8}$$

which can be solved for a and b as follows

$$a = \frac{N\sum_{i=1}^{N} x_i\, y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2} \tag{4.9}$$

$$b = \frac{\sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i\, y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2}. \tag{4.10}$$

Example 4.1.1: To illustrate how the least squares fitting is used, consider a transducer T3 with temperature measurements as described in Table 1:

Actual temperature (°F)	30	60	90	120	150	180
Output voltage (V)	0.90	1.64	2.30	2.90	3.60	4.31

Table 1 Temperature vs. Output Voltage for Transducer T3.

To compute the slope and offset that minimize the least squares error, Eq. (4.8) is used with the values in Table 2.

i	x_i	y_i	x_i^2	$x_i\, y_i$
1	0.90	30.00	0.81	27.00
2	1.64	60.00	2.69	98.40
3	2.30	90.00	5.29	207.00
4	2.90	120.00	8.41	348.00
5	3.60	150.00	12.96	540.00
6	4.31	180.00	18.58	775.80
Total	**15.65**	**630.00**	**48.74**	**1996.20**

Table 2 Summation Table for Example 4.1.1.

Equation (4.8) becomes

$$\begin{bmatrix} 48.74 & 15.65 \\ 15.65 & 6 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1996.2 \\ 630 \end{bmatrix} \tag{4.11}$$

which gives slope $a = 44.59$ and offset $b = -11.31$, and defines the line depicted in Figure 13. This line maps the transducer output voltage to the measured temperature minimizing the error relative to the actual value.

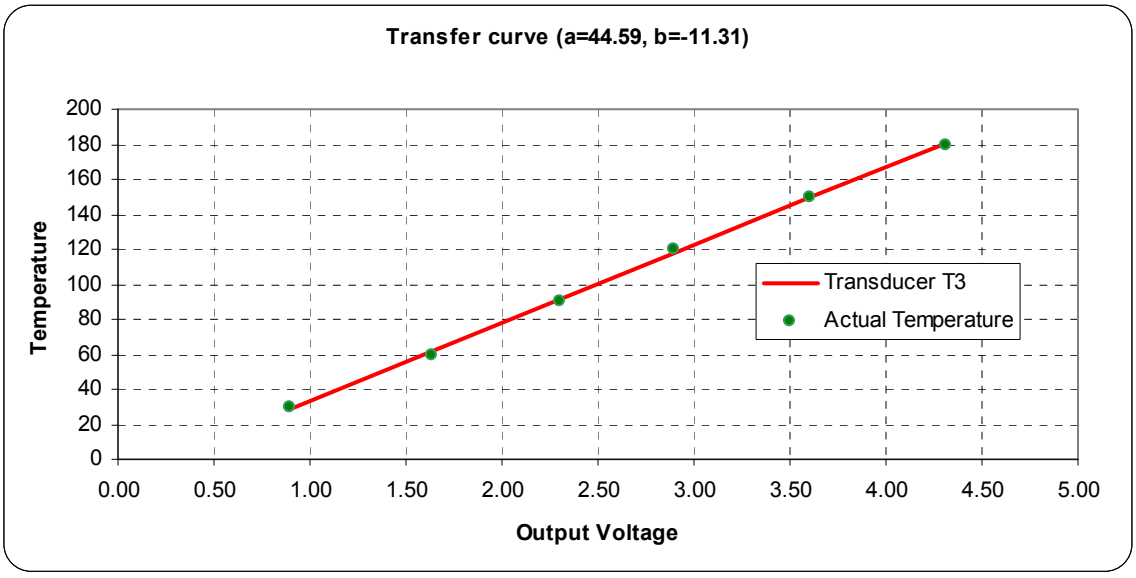

Figure 13 Transfer Curve After Linear Fitting in Example 4.1.1.

2. The Honeywell PPT Calibration Process

The Honeywell PPT calibration process can be easily understood with the help of Figure 14 which shows the calibration standard and a simplified model for the Honeywell PPT digital sensor. The figure illustrates the pressure readings from the calibration standard y_c and the readings from the digital sensor y_s, when both devices are measuring the same physical pressure p^*.

The digital sensor comprises three blocks:

- Block BT represents the piezoresistive transducer which converts the physical pressure p^* to an electric signal.
- Block B0 represents the default factory calibration and temperature compensation.
- Block B1 represents the compensation block in which constants a_1 and b_1 can be adjusted by the user.

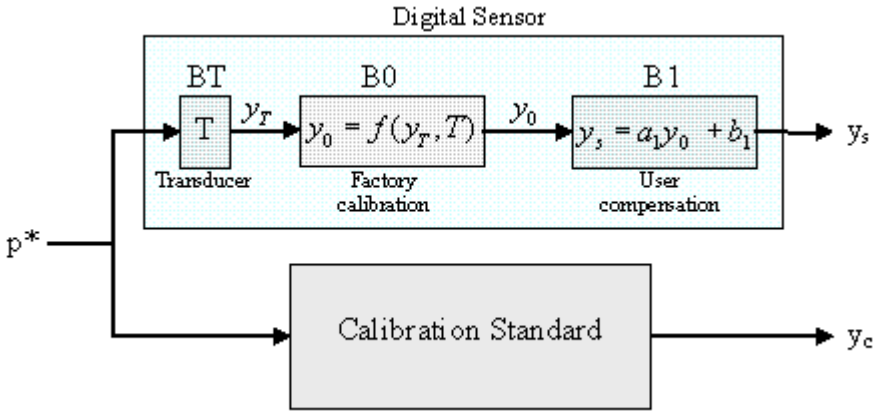

Figure 14 Introduction to Digital Sensor Calibration.

Similar to the previous examples, the goal of this calibration problem is to find a slope a and an offset b to replace a_1 and b_1, which minimize the least squares error between y_s and y_c. If y_0 were known, the values a_1 and b_1 would be computed by applying the Least Squares Fitting (LSF) method to a set of points (y_0, y_c). However, y_0 is not available. The sensor only provides the current slope a_1 and current offset b_1. Hence, the slope a and offset b that minimize the least squares error can be found using three different approaches described below.

a. Canceling the Current Compensation Block Prior to the Application of the LSF Method

Although the uncompensated value y_0 is not known, it can be found from y_s by canceling the compensation applied by block B1. Inverting $y_s = a_1 y_0 + b_1$ yields

$$y_0 = \frac{y_s - b_1}{a_1}. \tag{4.12}$$

Hence, the desired slope a and offset b can be found by just applying Eq. (4.8) to a set of points (y_0, y_c).

$$a = \frac{N\sum_{i=1}^{N}(y_0)_i\,(y_c)_i - \sum_{i=1}^{N}(y_0)_i \sum_{i=1}^{N}(y_c)_i}{N\sum_{i=1}^{N}(y_0)_i^2 - \left(\sum_{i=1}^{N}(y_0)_i\right)^2} \tag{4.13}$$

$$b = \frac{\sum_{i=1}^{N}(y_0)_i^2 \sum_{i=1}^{N}(y_c)_i - \sum_{i=1}^{N}(y_0)_i \sum_{i=1}^{N}(y_0)_i\,(y_c)_i}{N\sum_{i=1}^{N}(y_0)_i^2 - \left(\sum_{i=1}^{N}(y_0)_i\right)^2}. \tag{4.14}$$

b. Applying the LSF to a Set of Points (y_c, y_s) to Obtain a_2 and b_2 ($y_s = a_2 y_c + b_2$) in Order to Compute a and b

In this approach, the LSF is applied to a set of points (y_c, y_s) to find a linear relationship between the readings from the calibration standard and the sensor ($y_s = a_2 y_c + b_2$). Then, the slope a_2 and offset b_2, together with the current values of a_1 and b_1, are used to calculate the desired slope a and offset b. From Eqs. (4.9) and (4.10), a_2 and b_2 are computed as follows:

$$a_2 = \frac{N\sum_{i=1}^{N}(y_c)_i\,(y_s)_i - \sum_{i=1}^{N}(y_c)_i \sum_{i=1}^{N}(y_s)_i}{N\sum_{i=1}^{N}(y_c)_i^2 - \left(\sum_{i=1}^{N}(y_c)_i\right)^2} \tag{4.15}$$

$$b_2 = \frac{\sum_{i=1}^{N}(y_c)_i^2 \sum_{i=1}^{N}(y_s)_i - \sum_{i=1}^{N}(y_c)_i \sum_{i=1}^{N}(y_c)_i\,(y_s)_i}{N\sum_{i=1}^{N}(y_c)_i^2 - \left(\sum_{i=1}^{N}(y_c)_i\right)^2}. \tag{4.16}$$

The desired slope *a* and offset *b* can be found as follows:

$$\left.\begin{aligned} y_s &= a_1 y_0 + b_1 \\ y_s &= a_2 y_c + b_2 \end{aligned}\right\} \Rightarrow a_1 y_0 + b_1 = a_2 y_c + b_2 . \tag{4.17}$$

Rearranging the expression to obtain y_c as a function of y_0 yields

$$y_c = \frac{a_1}{a_2} y_0 + \frac{b_1 - b_2}{a_2} = a\ y_0 + b \tag{4.18}$$

which gives

$$a = \frac{a_1}{a_2} \tag{4.19}$$

$$b = \frac{b_1 - b_2}{a_2} . \tag{4.20}$$

c. Applying the LSF to the Readings (y_s, y_c) to Obtain a_3 and b_3 ($y_c = a_3 y_s + b_3$) in Order to Compute a and b

This approach is very similar to the previous one, however, the linear relationship is inverted ($y_c = a_3 y_s + b_3$). Applying the LSF to a set of points (y_s, y_c) yields:

$$a_3 = \frac{N\sum_{i=1}^{N}(y_s)_i (y_c)_i - \sum_{i=1}^{N}(y_s)_i \sum_{i=1}^{N}(y_c)_i}{N\sum_{i=1}^{N}(y_s)_i^2 - \left(\sum_{i=1}^{N}(y_s)_i\right)^2} \tag{4.21}$$

$$b_3 = \frac{\sum_{i=1}^{N}(y_s)_i^2 \sum_{i=1}^{N}(y_c)_i - \sum_{i=1}^{N}(y_s)_i \sum_{i=1}^{N}(y_s)_i (y_c)_i}{N\sum_{i=1}^{N}(y_s)_i^2 - \left(\sum_{i=1}^{N}(y_s)_i\right)^2} . \tag{4.22}$$

The desired slope *a* and offset *b* can be found as follows:

$$\left.\begin{aligned} y_s &= a_1 y_0 + b_1 \\ y_c &= a_3 y_s + b_3 \end{aligned}\right\} \Rightarrow y_c = a_3\left(a_1 y_0 + b_1\right) + b_3 . \tag{4.23}$$

Rearranging the expression to obtain y_c as a function of y_0 yields

$$y_c = a_3 a_1 y_0 \; + a_3 b_1 + b_3 = a \; y_0 + b \tag{4.24}$$

which gives

$$a = a_3 a_1 \tag{4.25}$$

$$b = a_3 b_1 + b_3 \,. \tag{4.26}$$

Comparing the equations $y_s = a_2 y_c + b_2$ and $y_c = a_3 y_s + b_3$, it is easy to show that $a_3 = 1/a_2$ and $b_3 = -b_2 / a_2$.

The desired slope *a* and offset *b* can be calculated using one of these three approaches just discussed. The calibration VI (Calibration.vi) uses the second approach because the application plots the fitting line each time the user acquires a new point during the calibration process. The second approach is better for this purpose because the plot is done with the readings from the calibration standard over the horizontal axis.

Having the desired slope *a* and offset *b*, more work is not needed if the digital sensor allows the replacement of a_1 and b_1 with the new calibrated values. However, that is not the case for the Honeywell PPT.

Block B0 of the Honeywell PPT does the temperature compensation with high accuracy. Hence, when the PPT needs to be calibrated, in general, the required adjustment on (a_1, b_1) is very small. For this reason, the PPT allows only small changes in these constants.

The default value for a_1 is 1 (one) and it is allowed to have only 0.6% variation from 0.994 to 1.006. The default value for b_1 is 0 (zero) and it can be adjusted only to values in the range from –0.6%FS to +0.6FS where FS is the sensor's full scale. Furthermore, a_1 and b_1 can only be modified by the addition of a multiple of 0.00005 or 0.00005*FS respectively:

$$a_1 = 1 + X \cdot 0.00005 \tag{4.27}$$

$$b_1 = 0 + Z \cdot 0.00005 \cdot FS \tag{4.28}$$

where X (slope adjustment) and Z (offset adjustment) are integers in the range $[-120, +120]$. Hence, the transfer curve $y_s = a_1 y_0 + b_1$ for the Honeywell PPT compensation block (a_1, b_1) is better written as:

$$y_s = (1 + X \cdot 0.00005) \cdot y_0 + Z \cdot 0.00005 \cdot FS \,. \tag{4.29}$$

The Calibration.vi was developed to perform the calibration of the Honeywell PPT by adjusting the slope and offset in Eq. (4.29). This task is accomplished by the use of the commands X=, Y=, Z=, and F= described below, available for the Honeywell PPT.

- The X= command adjusts the slope of the pressure output curve for positive pressures. It modifies the positive full scale slope of the PPT.
- The Y= command adjusts the negative full scale slope of differential PPTs.
- The Z= command adjusts the offset of the pressure output curve.
- The range of adjustments for X=, Y=, and Z=, commands is ±0.6%FS in 0.005% increments.
- The F= command can change the full-scale pressure span to any value between 50% and 100% of the factory specified range.

The X= and Y= commands make the slope adjustment, and the Z= command makes the offset adjustment. The plots on the left and right hand sides of Figure 15 illustrate the range in which the slope a_1 and the offset b_1 respectively are allowed to be adjusted.

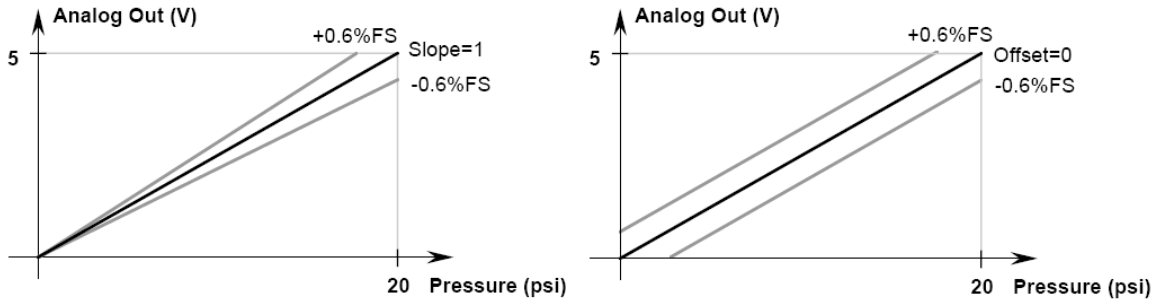

Figure 15 Custom Slope and Offset Options [From Ref. 3].

3. A Digital Sensor Calibration Example

To illustrate how the Calibration.vi works, consider the following example.

Example 4.3.1: Compute the new slope and offset adjustments to compensate the measurements displayed in Table 3 for a Honeywell PPT which has current slope and offset adjustments set to Xc = 120 and Zc = –120. Consider a full scale of 300 psi.

Calibrator	7.82	15.95	24.15	32.20	39.87
Sensor	5.92	14.08	22.36	30.44	38.17

Table 3 Data for the Honeywell PPT Calibration Example 4.3.1.

Figure 16 shows the Front Panel of the Calibration.vi after the acquisition of the points listed in Table 3. The results of the calibration computations are displayed in the bottom right corner. The white line shows the result of the Least Squares Fitting method applied to the measurements. Note that the white line fits the points which represent the sensor readings and is below the calibration line.

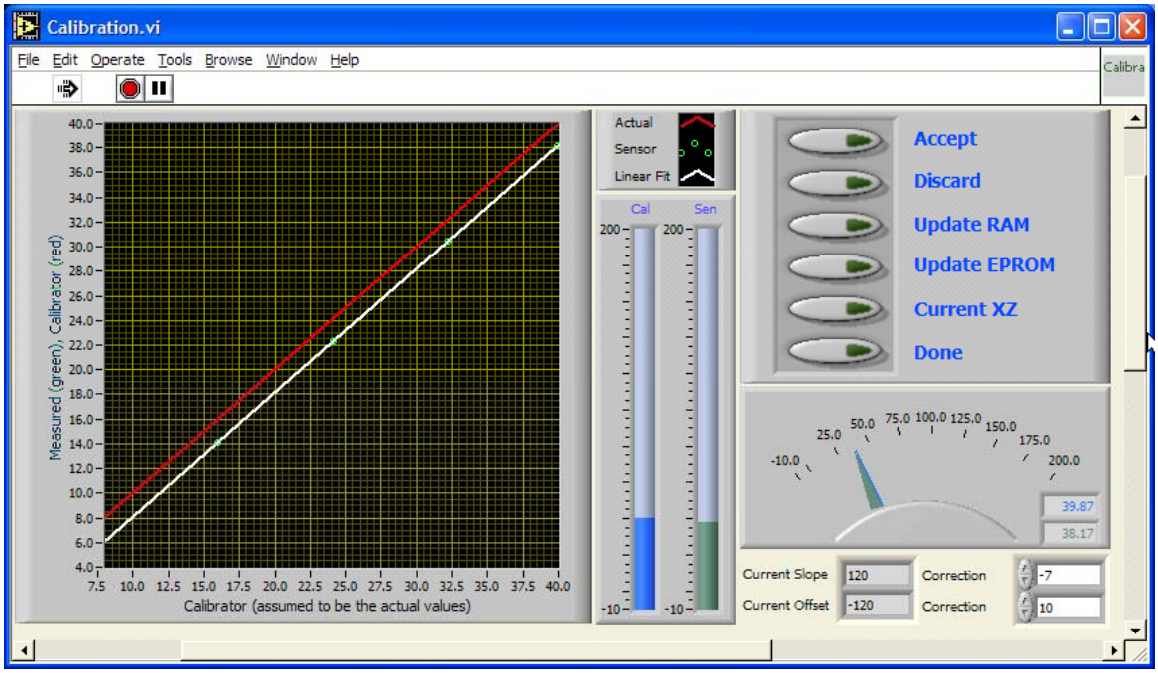

Figure 16 Calibration.vi During Honeywell PPT Calibration Example 4.3.1

a. Solution Using the First Approach

- Find the current compensation slope and offset (a_1, b_1) with FS = 300psi

  From Eq. (4.27): $a_1 = 1 + X_c * 0.00005 = 1.006$

  From Eq. (4.28): $b_1 = Z_c * 0.00005 * FS = -1.8$

- Find the uncompensated values y_0 from the sensor readings y_s

Sensor readings (y_s)	5.9200	14.0800	22.3600	30.4400	38.1700
$(y_s - b_1)/a_1$ **(y_0)**	7.6740	15.7853	24.0159	32.0477	39.7316

Table 4 Conversion Table for Example 4.3.1.a

- Find the desired slope a and offset b by linearly fitting the points (y_0, y_c)

  The summations required by Eq. (4.8) for the linear fitting are listed in Table 5.

	y_0	y_c	y_o^2	$y_0 y_c$
i	(x_i)	(y_i)	(x_i^2)	($x_i y_i$)
1	7.67	7.82	58.89	60.01
2	15.79	15.95	249.18	251.78
3	24.02	24.15	576.76	579.98
4	32.05	32.20	1027.06	1031.94
5	39.73	39.87	1578.60	1584.10
Total	**119.25**	**119.99**	**3490.49**	**3507.81**

Table 5 Summation Table for Example 4.3.1.a.

Equation (4.8) becomes

$$\begin{bmatrix} 3490.49 & 119.25 \\ 119.25 & 5 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 3507.81 \\ 119.99 \end{bmatrix} \tag{4.30}$$

which yields slope $a = 0.9997$, and offset $b = 0.1553$.

- Computing the new slope and offset adjustments

  $X_{new} = round((a-1)/0.00005) = -6$

  $Z_{new} = round(b/(0.00005 * FS)) = 10$

b. Solution Using the Second Approach

- Find the current compensation slope and offset (a_1, b_1) with FS = 300psi

  From Eq. (4.27): $a_1 = 1 + X_c * 0.00005 = 1.006$

  From Eq. (4.28): $b_1 = Z_c * 0.00005 * FS = -1.8$

- Find a_2 and b_2 by linearly fitting the measured values ($y_s = a_2 y_c + b_2$)

  The summations required by Eq. (4.8) for the linear fitting are listed in Table 6.

	y_c	y_s	y_c^2	$y_c y_s$
i	(x_i)	(y_i)	(x_i^2)	($x_i y_i$)
1	7.82	5.92	61.15	46.29
2	15.95	14.08	254.40	224.58
3	24.15	22.36	583.22	539.99
4	32.20	30.44	1036.84	980.17
5	39.87	38.17	1589.62	1521.84
Total	**119.99**	**110.97**	**3525.23**	**3312.87**

Table 6 Summation Table for Example 4.3.1.b.

Equation (4.8) becomes

$$\begin{bmatrix} 3525.23 & 119.99 \\ 119.99 & 5 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} 3312.87 \\ 110.97 \end{bmatrix} \tag{4.31}$$

which gives slope $a_2 = 1.0063$, and offset $b_2 = -1.9563$.

- Finding new compensation slope and offset

  From Eq. (4.19): $a = a_1 / a_2 = 0.9997$

  From Eq. (4.20): $b = (b_1 - b_2) / a_2 = 0.15532$

- Computing the new slope and offset adjustments

  $X_{new} = round((a-1)/0.00005) = -6$

  $Z_{new} = round(b/(0.00005 * FS)) = 10$

c. Solution Using the Third Approach

- Find the current compensation slope and offset (a_1, b_1) with FS = 300psi

  From Eq. (4.27): $a_1 = 1 + X_c * 0.00005 = 1.006$

  From Eq. (4.28): $b_1 = Z_c * 0.00005 * FS = -1.8$

- Find a_3 and b_3 by linearly fitting the measured values ($y_c = a_3 y_s + b_3$)

  The summations required by Eq. (4.8) for the linear fitting are listed in Table 7.

	y_s	y_c	y_s^2	$y_s y_c$
i	(x_i)	(y_i)	(x_i^2)	($x_i y_i$)
1	5.92	7.82	35.05	46.29
2	14.08	15.95	198.25	224.58
3	22.36	24.15	499.97	539.99
4	30.44	32.20	926.59	980.17
5	38.17	39.87	1456.95	1521.84
Total	**110.97**	**119.99**	**3116.80**	**3312.87**

Table 7 Summation Table for Example 4.3.1.c.

Equation (4.8) becomes

$$\begin{bmatrix} 3312.87 & 119.99 \\ 119.99 & 5 \end{bmatrix} \begin{bmatrix} a_3 \\ b_3 \end{bmatrix} = \begin{bmatrix} 3312.87 \\ 110.97 \end{bmatrix} \tag{4.32}$$

which gives slope $a_3 = 0.9937$, and offset $b_3 = 1.9440$.

- Finding new compensation slope and offset

  From Eq. (4.25): $a = a_1 * a_3 = 0.9997$

  From Eq. (4.26): $b = a_3 * b_1 + b_3 = 0.1553$

- Computing the new slope and offset adjustments

  $X_{new} = round((a-1)/0.00005) = -6$

  $Z_{new} = round(b/(0.00005 * FS)) = 10$

These news adjustments were also computed by the calibration VI resulting in $X_{new} = -7$ and $Z_{new} = 10$, as shown in Figure 16. The small difference in the slope adjustment is due to rounding errors in Table 3 and subsequent computation.

Figure 17 shows the readings from the sensor (red line) and the calibration standard (white line) prior to the execution of the calibration process. The difference between the readings is almost two psi.

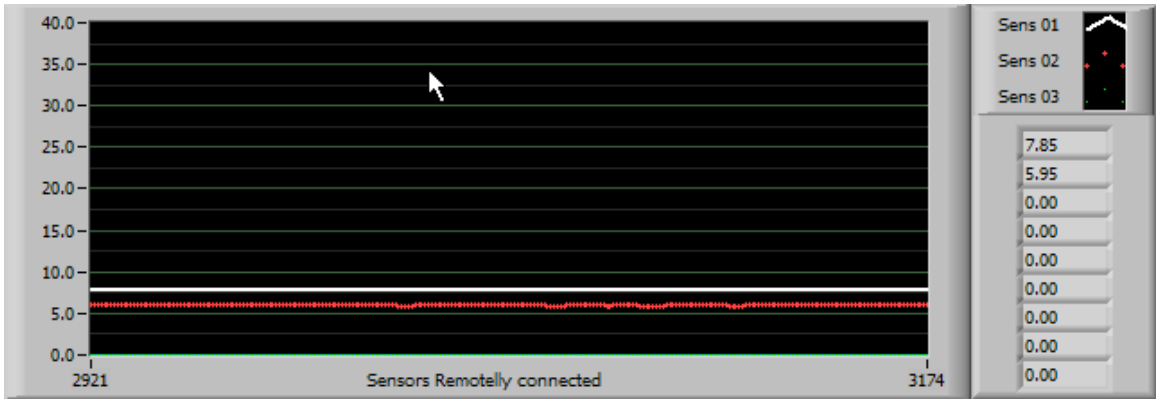

Figure 17 Sensor and Calibration Standard before Calibration.

The new readings after the calibration process are shown in Figure 18 with an actual pressure of 5.14 psi and in Figure 19 with an actual pressure of 39.57 psi. Notice that in both figures the lines overlapped demonstrating that the calibration is effective through the entire range over which the calibration is made.

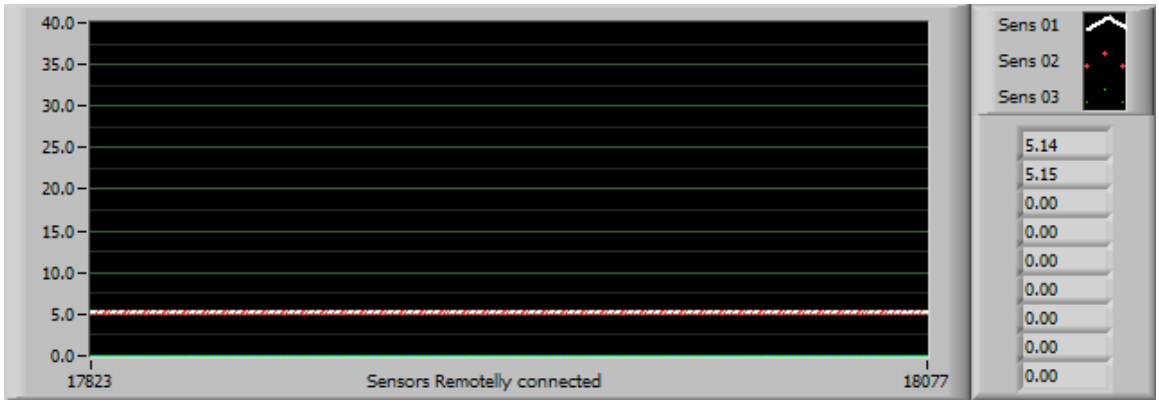

Figure 18 Sensor and Calibration Standard after Calibration when the Actual Pressure is 5.14 psi.

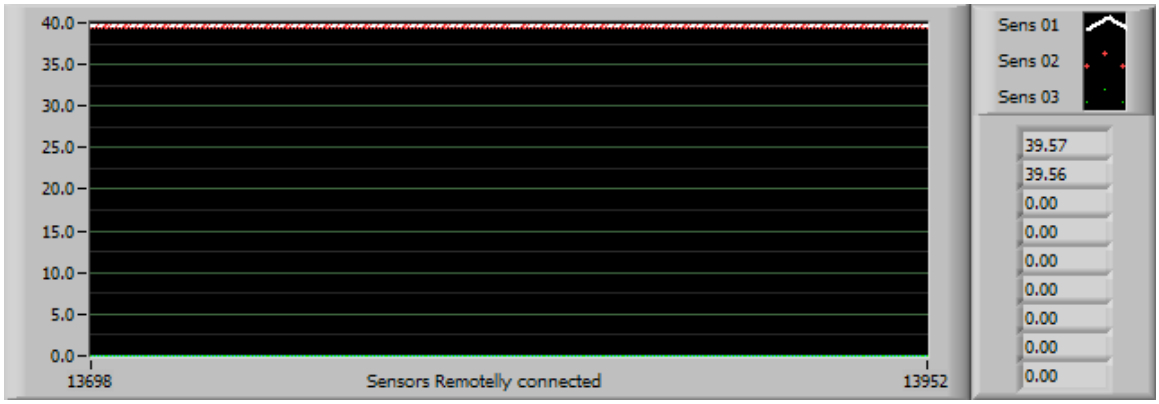

Figure 19 Sensor and Calibration Standard after Calibration when the Actual Pressure is 39.57 psi.

E. MAIN SUBVIS OVERVIEW

The Master Module hierarchy is summarized in Figure 20. Since some of the VI's used by this application are well-known from the LabVIEW libraries and LabVIEW examples, they are not discussed here. The main VI's specifically designed and developed for this application are listed in Table 8 and briefly discussed below.

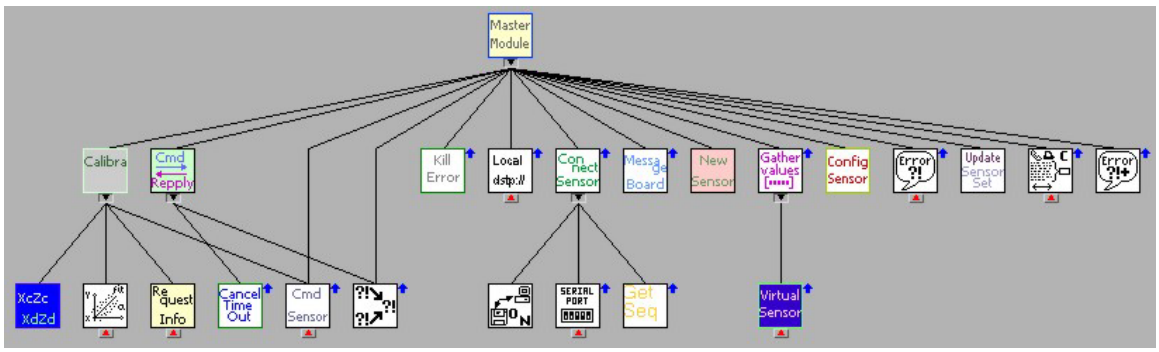

Figure 20 Master Module sub-VIs Hierarchy.

VI	Brief Discussion
	This VI starts the Honeywell PPT calibration process where the user can continuously acquire pressure measurements. During the acquisition, the VI does the linear fitting of the sensor's points with respect to the calibration points (assumed to be the actual pressure). Hence, considering the calibrator as a reference, the sensor's accuracy is displayed graphically and numerically in terms of slope and offset. The process also allows the user to store the computed corrections to the sensor RAM or EEPROM at any time.
	This VI uses the datasocket command channel to send commands to the Slave Module and waits for a reply. If an answer is not received after a time out period, an error is generated.
	This VI takes the error cluster and an array of error codes to be ignored and gives an error cluster free of the requested errors. It also signalizes the errors found.
	RequestNewSensorInfo.vi displays a modal dialog box where the user can define a new sensor to be connected and monitored or calibrated. A cluster with the user information is given as output.
	This VI takes the cluster with the user information, tries to establish a connection and gives the complete sensor cluster with the information required for the monitoring and calibration process. If the attempt to establish a connection fails, an error message is generated.
	This VI is used in the Master Module and Slave Module to gather the measurements of the sensors connected to the respective module. It takes the Sensor Set array and gives an array of doubles comprising the values.
	This VI updates the Main Module Sensor Set with the copy received from the Slave Module. The Slave Module always replies with a copy of its Sensor Set when it receives a command that causes a modification of any of the sensors' fields.

VI	Brief Discussion
	This VI is called by the calibration VI to request information from the sensors. It can request the current slope, current offset or even the current measurement.
	This VI sends commands to a sensor if the sensor is connected to the module from which this VI is being called, and updates the Sensor Set accordingly.
	This VI takes a sensor command, the sensor family and the sensor mode as input and produces an array of strings comprising the sensor language strings needed to perform the given command.
	This VI takes the information needed to identify a sensor in the sensor set, the sensor set itself and the command string. It sends the command to the actual sensor, reads the reply, parses the measurement and provides the numerical value.
	This VI takes a TCP connection reference number, a regular expression to identify the substring to be parsed and provides the last match found in the TCP buffer.
	This VI takes a UDP connection reference number, a regular expression to identify the substring to be parsed and provides the last match found in the UDP buffer.
	This VI is used by the virtual sensor VI to communicate to local sensors connected through serial ports.
	This VI takes a TCP connection reference number and a string and sends the string through the TCP connection.
	This VI takes a UDP connection reference number and a string and sends the string through the UDP connection.

Table 8 Summary of the Key VI's Designed for the Main Module.

F. COMMUNICATION THROUGH THE COMMAND-REPLY CHANNEL

As stated previously, the application uses DataSocket to establish two different communication channels between the Master and Slave modules. The first channel, named "data channel", has only one direction and is used to receive data from remotely connected sensors. On the Slave Module side of this channel, there is one dedicated thread which continuously reads the remote sensors' data, organizes the data sequentially in one numerical array and sends the array to the Master Module, which may or may not receive the information. If the data array is not received, the Master Module assumes that the last reading is still valid. Hence, the "data channel" does not need strong mechanisms to guarantee data delivery because the continuous data flux ensures that the lost information will be updated in the next cycle.

The second channel, named "command-reply channel", is a two-way communication channel used to transmit commands from the Master to the Slave Module and replies in the opposite direction. When the Master Module sends a command to the Slave Module, it has to be sure that the requested actions were really carried out because both modules need to keep their sensor tables synchronized. The sensor table or sensor set holds the status of all connected sensors and should be updated in both modules when the user changes the number of connected sensors or the status of a particular sensor.

The subVI CommandReply.vi, whose diagram is depicted in Figure 21, was created with two main goals. The first is to let the Master Module detect if the command was received and accomplished and the second is to allow the Slave Module to return a reply when necessary.

When the command cannot be performed, the Slave Module returns an error message. If a reply is not received during the expected time, the subVI generates a time out error.

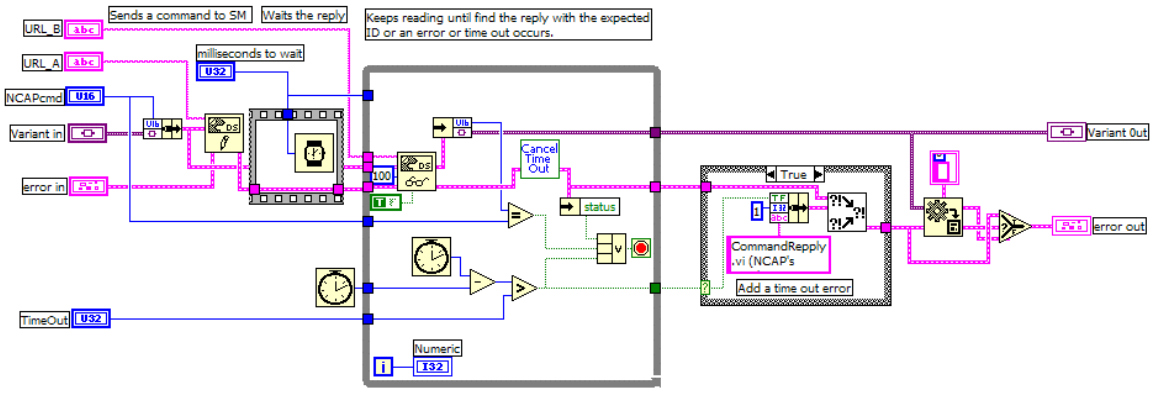

Figure 21 Diagram of CommandReply.vi.

The command-reply channel is implemented using a DataSocket connection for each direction. Thus, the command-reply VI requires two URLs as input. URL_A defines the DataSocket connection used to write commands, while URL_B defines the DataSocket connection used to read replies.

The Slave Module is able to understand 19 commands previously defined as listed in Table 9. Each command has a code, the number of arguments and the type of each argument. Since each command can have arguments with different data types, the command-reply VI uses a variant type for the argument input terminal.

When the Master Module sends a command through the command-reply VI, the command code (unsigned integer) and the variant are bundled in a cluster prior to the transmission. Hence, when the cluster reaches the destination, the Slave Module needs to decode it to ascertain the original data type of each argument.

After command and arguments are decoded, the Slave Module performs the requested actions and always replies with the same command code. If the command is a query for some information, the reply comprises the command code and the desired information. If an error occurs during the command evaluation, the reply comprises the command code and an error cluster with the error code and respective message. If none of the two conditions above occurs, the reply comprises the command code and an empty variant.

Code	Command	Argument type	Reply's data type
01	useContinuousMode	Unsigned Integer	- o -
02	useSingleMode	Unsigned Integer	- o -
03	measurePressure	Unsigned Integer	- o -
04	measureFahrenheit	Unsigned Integer	- o -
05	measureCelsius	Unsigned Integer	- o -
06	selectTargetSensor	Unsigned Integer	- o -
07	getSensorMap	- o -	cluster
08	setMonitoredStatus	Unsigned Integer Boolean	- o -
09	getPressure1	Unsigned Integer	double
10	getXZ	Unsigned Integer	Array of Integers
11	setXZ	Unsigned Integer Array of Integers	- o -
12	save2EEPROM	Unsigned Integer	- o -
13	addSensor	Cluster	- o -
14	removeSensor	Unsigned Integer	- o -
15	selectCalibrator	Unsigned Integer	- o -
16	getPressure2	Array of UI	double
17	getSlope	Array of UI	Integer
18	getOffset	Array of UI	Integer
19	matchSensors	Array of UI	Boolean

Table 9 List of Slave Module's Commands.

G. SUMMARY

In this chapter, the major features of the Master Module were presented with emphasis on the multithread characteristics, the sensor connection process, the communication mechanism and the calibration process. Three approaches were provided for the computation of the calibration process and, for each approach, an example is given to show the details related to every step. The next chapter presents the Slave Module.

THIS PAGE INTENTIONALLY LEFT BLANK

V. THE SLAVE MODULE

This chapter presents the Slave Module and gives a brief description of its main subVIs. The emphasis is given to the SlaveCore.vi that plays a central role in the Slave Module.

A. INTRODUCTION

In contrast to the Master Module, the Slave Module (SM) does not have a Graphical User Interface (GUI) directly controlled by the user. However, the Slave Module performs an important role when an NCAP is used to gather the sensor data in a shipboard system. With both modules, the application can talk to the NCAP through a DataSocket communication channel. Hence, all the sensors connected to the NCAP through the Slave Module can be monitored and calibrated in the same manner discussed for the Master Module.

B. MULTI-THREAD FEATURES

During the connection process, all the sensors marked to be remotely connected stay under the Slave Module control until the sensor is removed. If the user wants to control the sensor from the Master Module, the sensor has to be reconnected. After initialization, the Slave Module starts one thread to gather measurements from all the sensors whose connections are under SM control. Hence, even without the direct control of those sensors, the Master Module continuously receives information from them.

A second thread is also started after initialization to ensure that the SM is always listening to the commands sent by the MM. When the SM receives a command, the SMcore.vi is called to perform the required actions and provide the MM with the expected reply or an error message.

The SM threads are depicted in Figure 22 and the subVI's hierarchy in Figure 23.

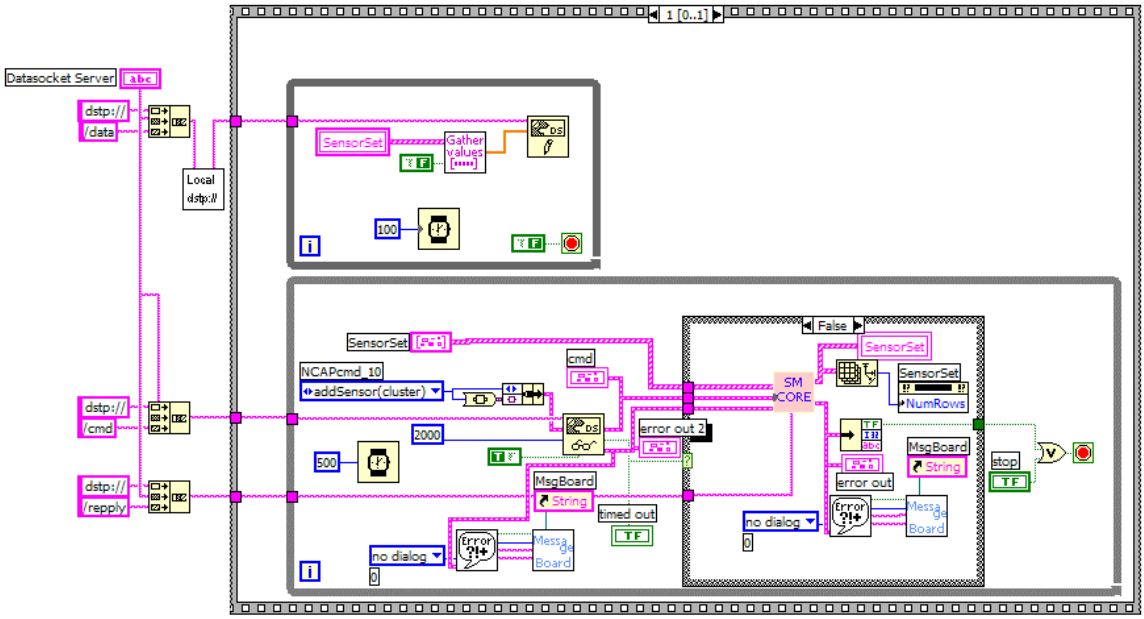

Figure 22 Threads Running in the Slave Module after Initialization.

C. MAIN SUBVIS OVERVIEW

The Slave Module high levels subVIs can be seen in Figure 23, where the VI's hierarchy is expanded two levels down. Some of the subVIs will not be discussed here because they were already discussed in the previous chapter or are well-known from the LabVIEW libraries and LabVIEW examples. The subVIs exclusively designed for the SM are briefly discussed below.

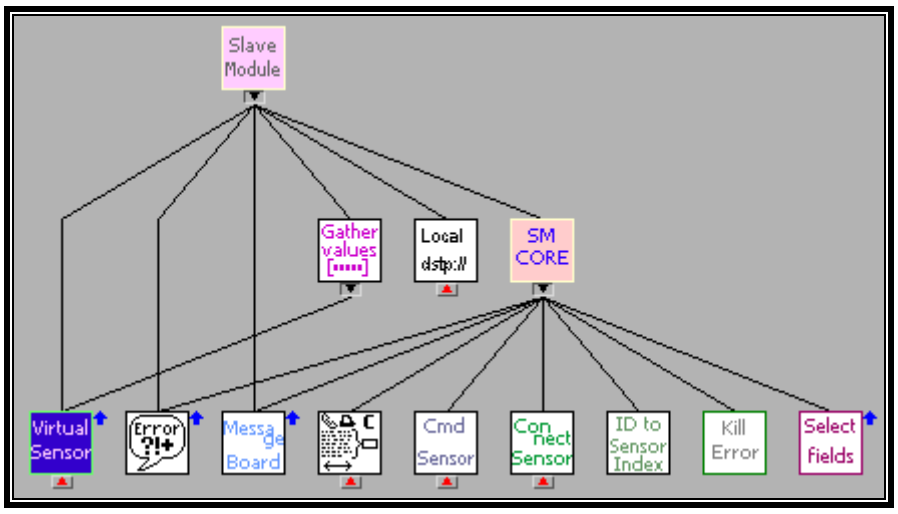

Figure 23 Slave Module Hierarchy Tree.

VI	Brief Discussion
	This VI is the main subVI in the SM; it takes the command sent by the MM, the reply URL, the Sensor Set and an error cluster as input. Performs the requested action, sends a reply to the MM and returns the updated Sensor Set and an error cluster as output.
	This VI is used when the entire Sensor Set needs to be sent through the DataSocket channel. It was developed to make a copy of the Sensor Set excluding the fields not supported by the DataSocket protocol.
	This VI takes a sensor ID and the Sensor Set and returns the sensor cluster and its index.

Table 10 Summary of subVIs Designed for the Slave Module.

The SlaveCore.vi subVI depicted in Figure 24 is the “heart” of the Slave Module and is responsible for executing the actions required by the commands received from the MM.

The diagram in Figure 24 shows how the “add sensor” command is implemented. It takes the current Sensor Set and a cluster with the command ID and a variant data. When the subVI is called, it first identifies the command in order to identify which type of data is being transported inside the variant element.

In the case of the “add sensor” command, the data required to perform the command also depends on which module the sensor is being connected. When the sensor is connected to the Master Module, the Slave Module receives a cluster with the information of an already connected sensor and only needs to update its Sensor Set. On the other hand, if the sensor is not yet connected and should be connected to the Slave Module, the cluster received contains the information provided by the user to establish the desired connection. Hence, it establishes a connection if necessary, updates the Sensor Set, and sends a modified copy of the updated Sensor Set, excluding elements not supported by DataSocket, as a reply to the Master Module.

The SlaveCore.vi is also able to perform commands to switch the sensor mode to single or continuous, to read pressure or temperature, to read or write slope and offset used in the calibration process, to set a sensor as a calibrator, and others.

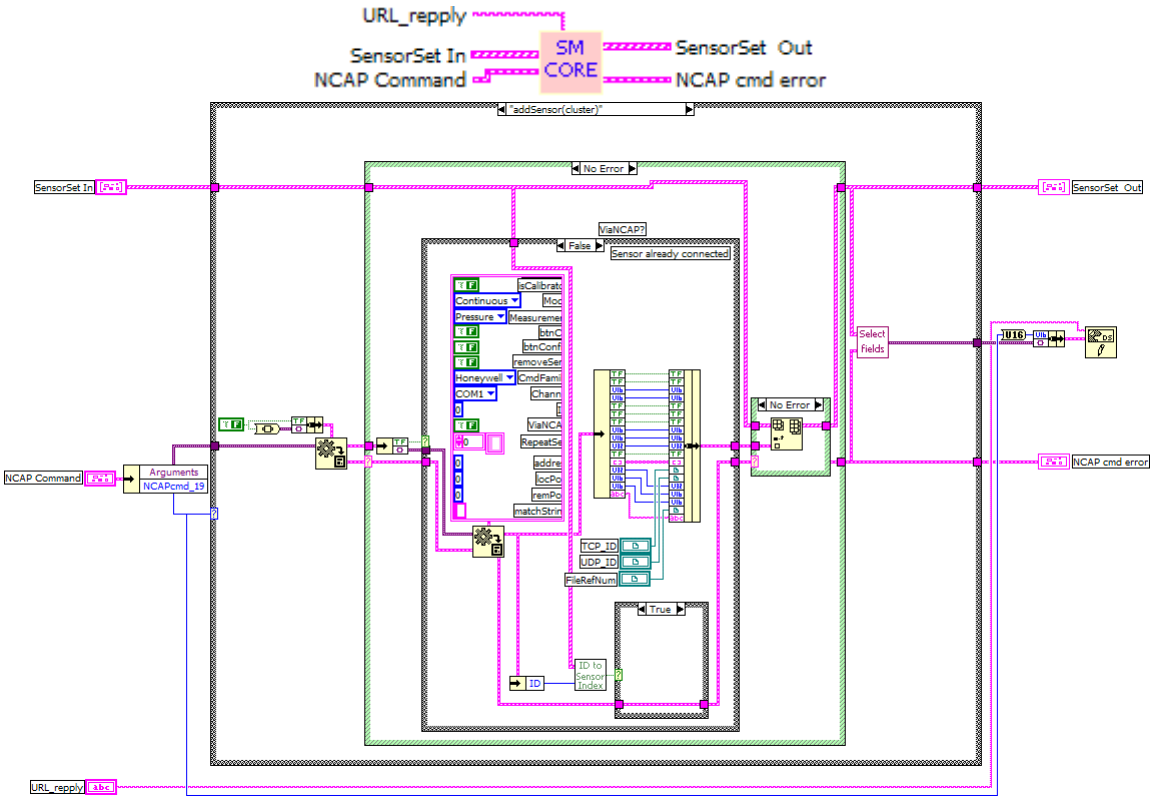

Figure 24 Slave Module Main sub-VI (SlaveCore.vi).

The subVI SelectFields.vi has its diagram depicted in Figure 25. It takes the Sensor Set and an error cluster as input and returns a variant as output. The variant will contain a modified copy of the Sensor Set or the input error, if an error is detected. The modified copy is necessary because the Sensor Set received as input contains reference numbers to the TCP and UDP connection and to the opened file. These elements cannot be sent through the DataSocket channel.

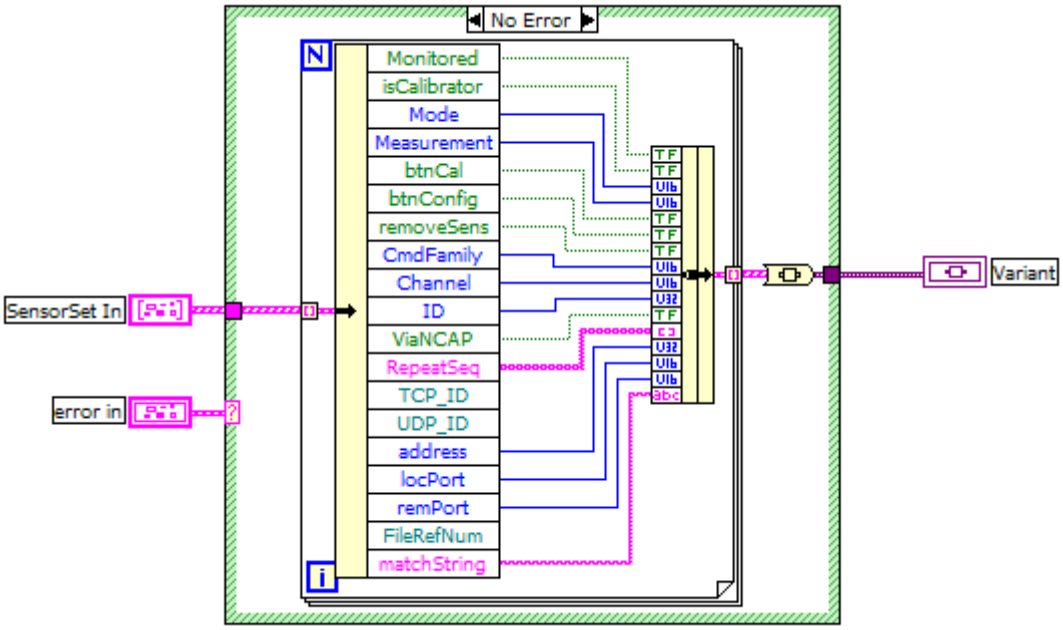

Figure 25 Diagram of SelectFields.vi.

The IDtoIndex.vi is a very simple subVI used to find a sensor in the Sensor Set. It takes the sensor ID and uses the while loop depicted in Figure 26 to find the sensor index. It also returns the matched sensor and a Boolean to indicate if the sensor was found.

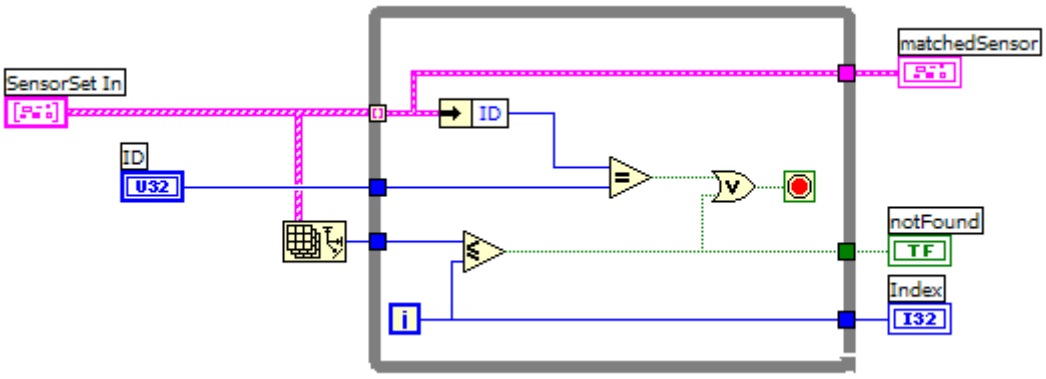

Figure 26 Diagram of IDtoIndex.vi.

D. SUMMARY

In this chapter, the Slave Module threads were presented with a brief description of its main subVIs. Emphasis is given to the SlaveCore.vi which is the main subVI. The next chapter presents a simplified version of the calibration system designed for the calibration of analog sensors.

VI. THE NEW CALIBRATION SYSTEM FOR ANALOG SENSORS

This chapter presents a simplified calibration system focusing on the monitoring and calibration of shipboard analog sensors. It offers the same calibration capabilities discussed in the previous chapters by means of a new GUI based on the prototype developed in [Ref. 5].

A. INTRODUCTION

Although the Master Module and Slave Module could be easily adapted to control, monitor and calibrate analog sensors, they were designed to handle only digital sensors. Rather than add one more capability to this application, the decision was made to create another application that is lighter and easier to use, which could be tested and evaluated by the Land-Based Testing Facility at NAVSEA-Philadelphia with a good chance of being approved for use onboard ships in the near future. The Analog.vi was developed with this purpose in mind.

To handle analog sensors, several features provided by the Master Module and Slave Module are not required. The connection process, the two-way communication channels established with each sensor, and other features offered by the dual module design are beyond the need of analog sensors and provide an extra level of complexity not required for the calibration of analog sensors in current shipboard systems.

B. THE MAIN VI AND THE USER INTERFACE

The prototype presented in [Ref. 5] is the starting point for this new calibration system for analog sensors.

The Analog.vi has three threads running on the operator's laptop/tablet PC depicted by the three loops in the block diagram of Figure 27. The central loop performs the main thread, which is responsible for reading data from three different origins: the analog sensor, the calibration standard and the ICAS. The upper loop is responsible for the detection of events generated by the user and the lower loop is responsible for starting the calibration process when requested by the user.

In the main thread, the analog sensor data comes from the NCAP through a DataSocket connection and is represented by a voltage between 1 and 5 V which is locally converted to the pressure measurement. The local variables “multiplier” and “additive” are used to store the slope and offset used to make this conversion.

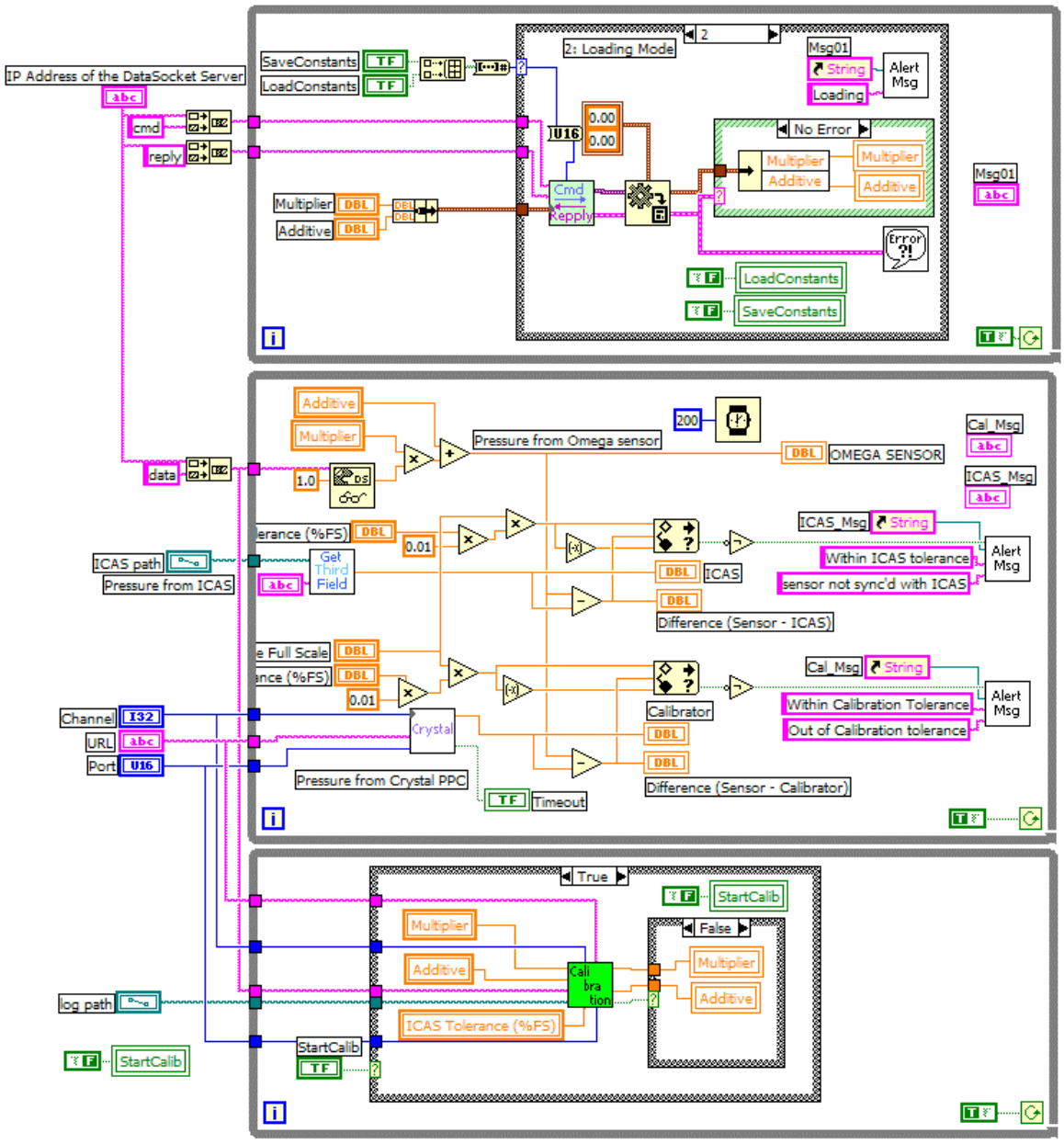

Figure 27 Block Diagram of the Analog.vi.

The Crystal PPC is also used by the Analog.vi as the calibration standard and can be wired or wirelessly connected to the laptop/tablet PC.

The ICAS readings are provided only for test and evaluation purposes. It allows the operator to detect if the local measurement matches the measurements taken by the Integrated Condition Assessment System (ICAS).

Figure 28 shows the application graphical user interface which comprises four sub-panels: the control sub-panel in the top left position, the sensor sub-panel in the top right position, the ICAS sub-panel in the bottom left position, and the calibration standard sub-panel in the bottom right position.

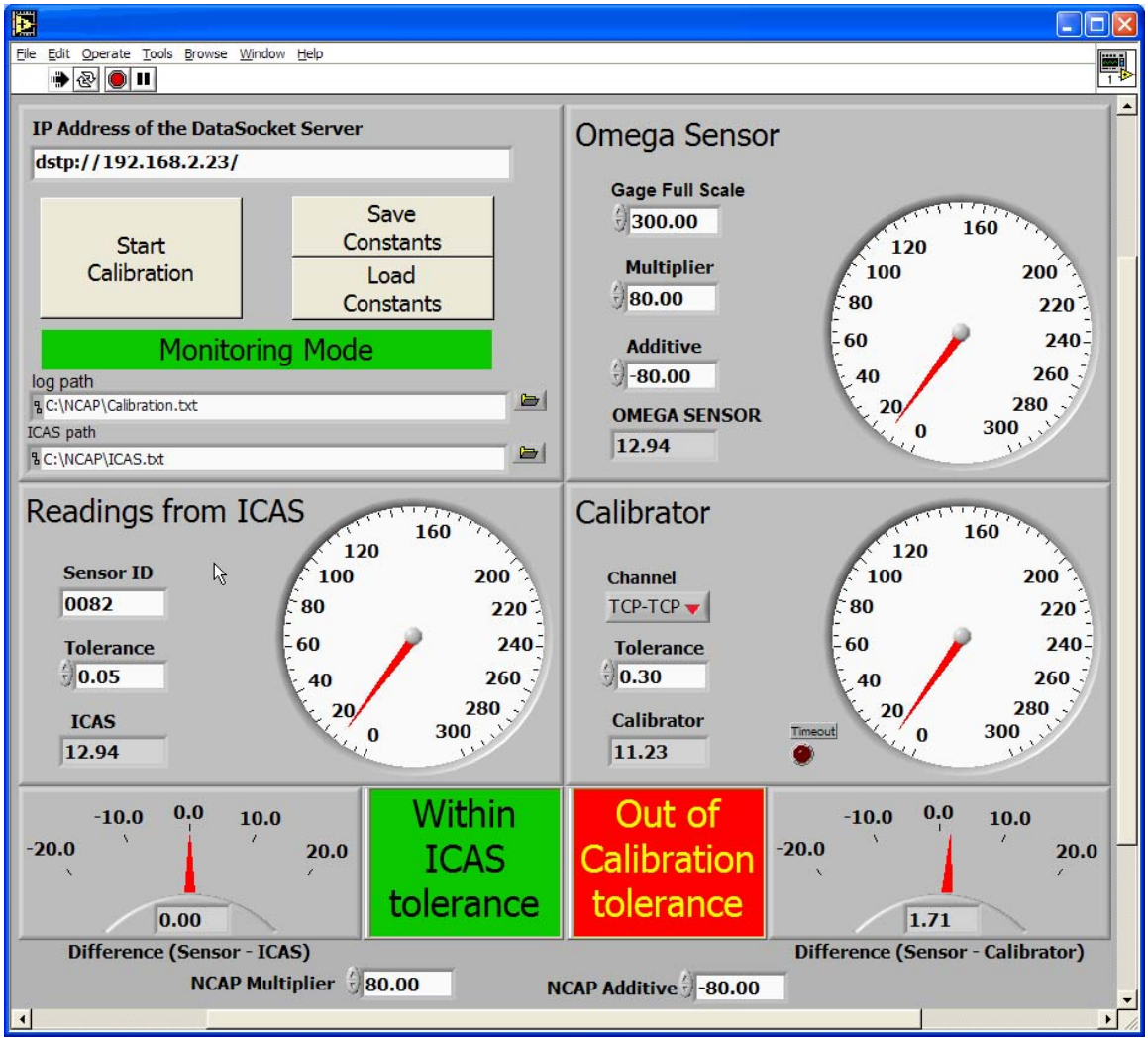

Figure 28 Front Panel of the Analog.vi.

The control sub-panel has textboxes used to provide the DataSocket Server address, the path to the calibration log file, and the path to the ICAS file. The Calibration.txt

file depicted in Figure 29 is used to maintain a history file of the results generated by the calibration subVI. The ICAS file depicted in Figure 30 is used, for evaluation purposes, as a communication medium between this application and the ICAS system.

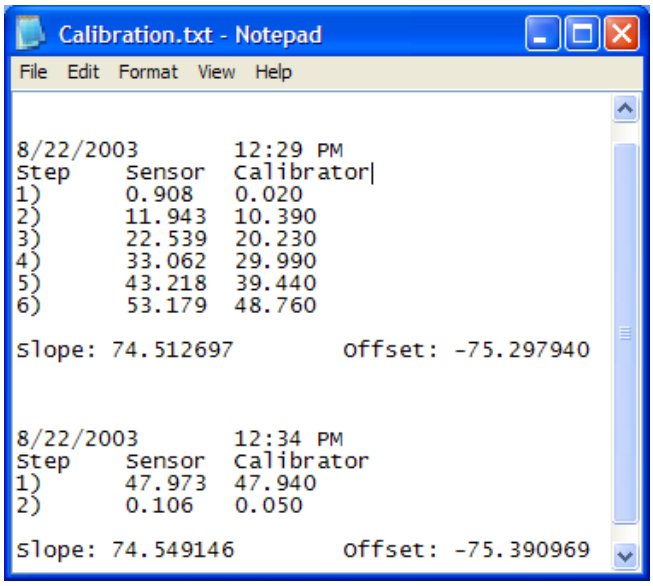

Figure 29 Log File Generated or Updated at the End of Calibration Process.

In the top right position, the sensor sub-panel has a gauge used to display the readings from the analog sensor. The first textbox provides the sensor full scale value which is used to compute the sensor absolute tolerance. The local variables "multiplier" denoted by a and "additive" denoted by b are used to convert the voltage read from the DataSocket connection x_s to the pressure value y_s.

$$y_s = ax_s + b\,. \tag{4.33}$$

In the bottom left position, the ICAS sub-panel has a gauge to display the readings from the ICAS system. For evaluation purposes, the agreement was to obtain the ICAS value through a text file which is updated by the ICAS system periodically. The ICAS updates the file by appending a line with several fields as shown in Figure 30. The line has the sensor ID in the first field, the pressure measurement (upon updating) in the third field, and the other fields are filled with information related to the sensor. In order to find the latest pressure written to the file for a specific sensor, a subVI was developed to perform a backward file search for the desired sensor. This subVI takes a sensor ID, finds

the latest appended line related to the desired sensor and parses the pressure value stored in the third field. This value is displayed and also compared to the sensor local readings to verify if they are synchronized by the given tolerance.

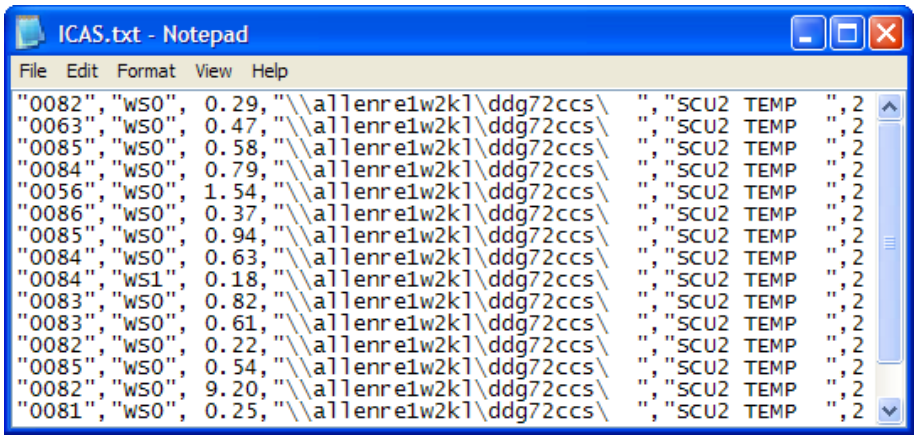
ICAS.txt - Notepad
File Edit Format View Help
"0082","WS0", 0.29,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0063","WS0", 0.47,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0085","WS0", 0.58,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0084","WS0", 0.79,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0056","WS0", 1.54,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0086","WS0", 0.37,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0085","WS0", 0.94,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0084","WS0", 0.63,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0084","WS1", 0.18,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0083","WS0", 0.82,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0083","WS0", 0.61,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0082","WS0", 0.22,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0085","WS0", 0.54,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0082","WS0", 9.20,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2
"0081","WS0", 0.25,"\\allenre1w2k1\ddg72ccs\ ","SCU2 TEMP ",2

Figure 30 The ICAS.txt File.

In the bottom right position, the calibration standard sub-panel displays the readings from the calibration standard (Crystal PPC). This sub-panel has a text box where the tolerance for the sensor readings can be specified and a menu ring used to define the type of connection used by the calibration standard, which can be RS232 cables, Bluetooth, or an 802.11b wireless LAN.

C. VIs RUNNING ON THE NCAP

On the NCAP side, two VIs are running, perchs_daq1_local.vi [Ref. 5] and HandleNCAPConsts.vi. Figure 31 depicts the diagram of the first one and is responsible for sending the analog sensor data through the DataSocket "data" connection. This single connection is used to provide a DataSocket communication channel from the NCAP to the user's laptop/tablet PC. Two others DataSocket connections, named "command" and "reply" connections, are used to implement a two-way communication channel to allow the management of the calibration constants in the NCAP configuration file.

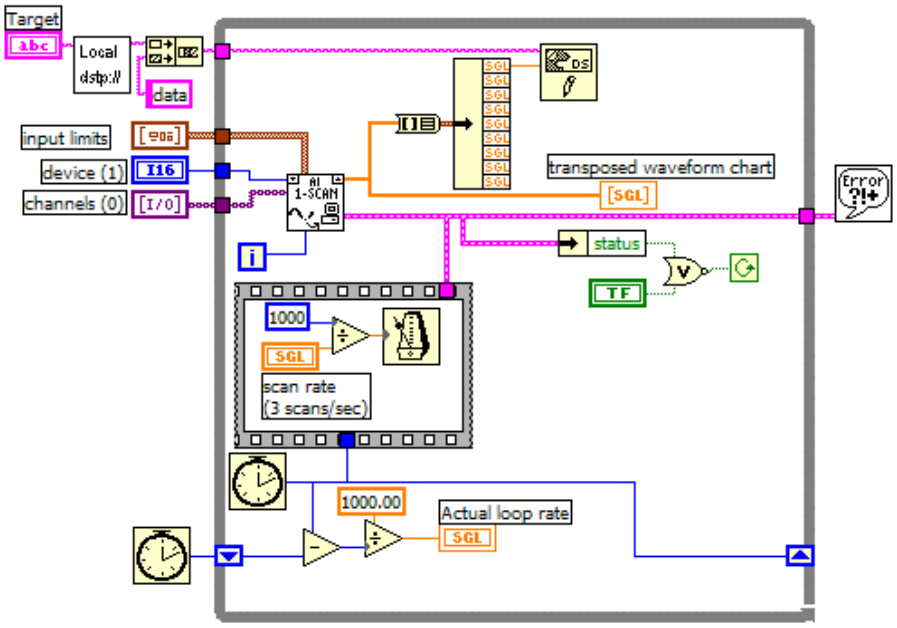

Figure 31 Diagram of perchs_daq1_local.vi [From Ref. 5].

Figure 32 depicts the HandleNCAPConsts.vi diagram which is responsible for receiving, executing and replying read or write commands. The read command is sent when the user wants to load the current slope and offset values stored in the file. The write command is typically sent after a calibration session when the slope and offset that minimizes the Least Squares Errors are computed and the user wants to update the configuration file.

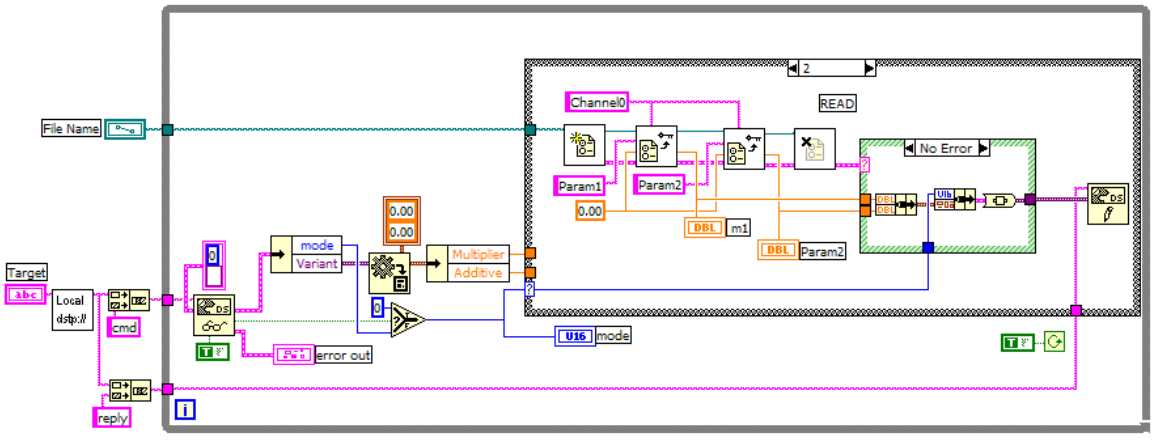

Figure 32 Diagram of HandleNCAPConsts.vi.

D. THE CALIBRATION PROCESS

In order to calibrate analog sensors, the OmegaCalib.vi was developed to allow the user to start an iterative process with three basic steps:

- read the directions displayed in the message board;
- adjust the pressure accordingly; and
- accept the measurements by clicking the OK button.

The three steps are repeated as many times as required by the current calibration procedures. Each time a new point is acquired, the subVI's Front Panel is appended with a new record comprising the calibration standard (PPC) readings, the sensor readings and the limits of the sensor tolerated range. Additionally, a graphical visualization of the measured values along with the actual pressure is provided. Figure 33 shows the user interface used during the calibration process. Note that the white line is fitted to the measured values and is above the line of the actual pressure values (red line). This situation indicates that the current slope and offset are generating sensor readings greater than the calibration standard readings. The Front Panel also shows the current slope and offset and the computed values required to minimize the Least Squares Errors.

The user can locally validate or abort the calibration process by pressing the "Done" or "Abort" buttons, respectively, at any time during the calibration process.

Figure 34 depicts the Block Diagram of the OmegaCalib.vi. It is composed of an initialization block, the main loop and the final block. The initialization block is used to set the controls, variables and indicators to initial values required by the main loop. The main loop is the block that interacts with the user and performs the calibration process. The final block is responsible for generating the log file depicted in Figure 29.

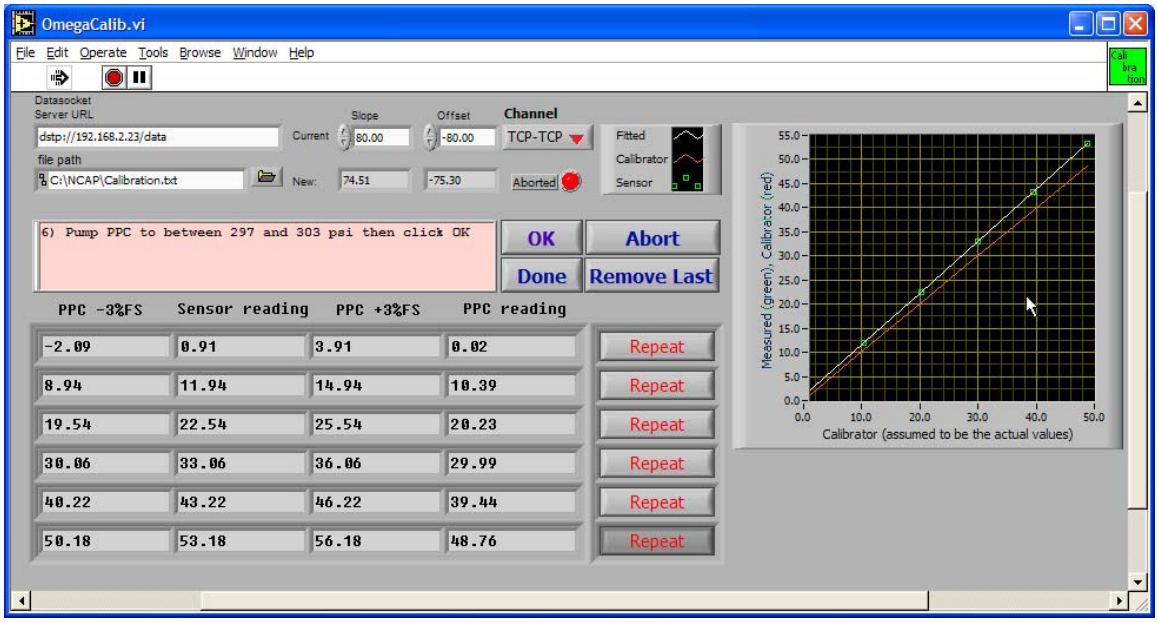

Figure 33 User Interface for the Calibration Process.

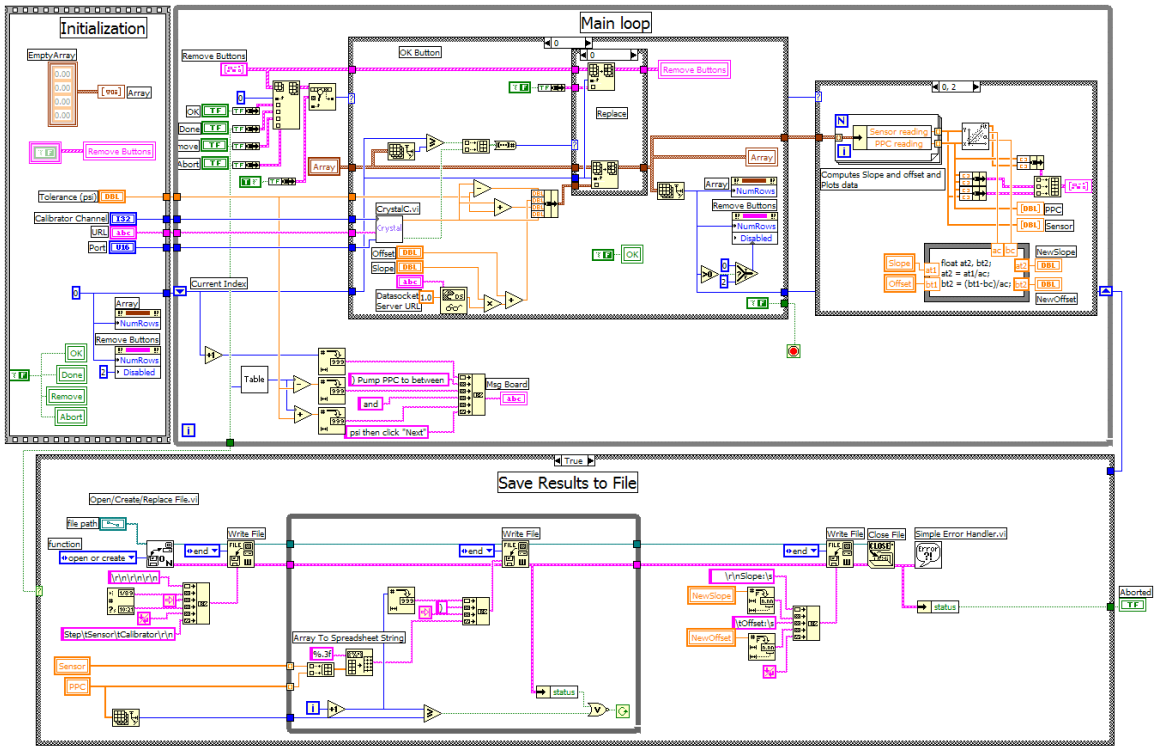

Figure 34 OmegaCalib.vi Block Diagram.

Figure 35 illustrates the Front Panel of the Analog.vi after the calibration process. Note that the local variables "multiplier" and "additive" were updated and that the error between the sensor and calibrator standard is very small.

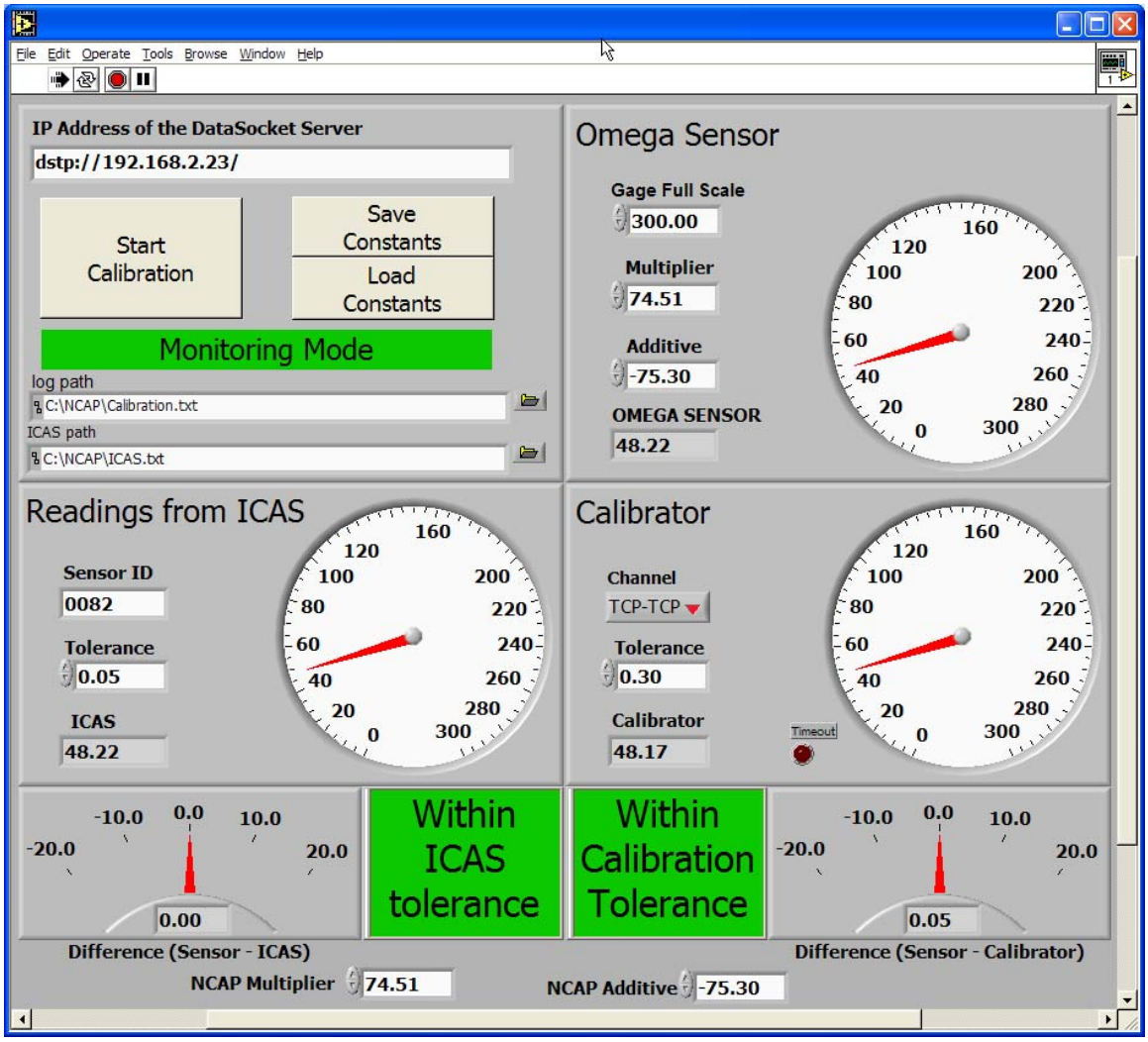

Figure 35 Front Panel of Analog.vi After the Calibration Process.

E. SUMMARY

In this chapter, a new calibration system was designed to perform the monitoring and calibration of analog sensors already used by shipboard systems. The GUI developed in [Ref. 5] was taken as the starting point to obtain a lighter version of the new calibration system. The next chapter presents the conclusions and explains the accomplishments of the thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS

A. SUMMARY

This thesis presented the design and development of a new closed-loop calibration system for shipboard pressure sensors. The software of the calibration system was implemented using LabVIEW, which is the industrial standard language for measurements and automation. The calibration system takes advantage of the latest wireless communication technologies including 802.11b wireless LAN and Bluetooth. The version of the calibration system developed at the completion of this thesis was operational and successfully demonstrated to the project sponsor.

The fundamental approach taken in this thesis was to wirelessly transmit both the sensor data and calibration standard to a Tablet PC. The Tablet PC compares pressure measurements from the sensor and calibration standard at multiple pressure points generated by a portable hand pump. The calibration constants (slope and offset) are automatically computed, and downloaded to the smart sensor at the request of the user.

The newly developed calibration system has the following features:

- It adopts a modular approach to software development, which makes it easy and flexible to use and expandable in the future. The system consists of two main modules named Master Module and Slave Module. The Master Module always resides on the Tablet PC, providing a graphical user interface (GUI) for the user to enter commands and view measurements and other useful information. The Slave Module can reside on the same Tablet PC, or on a remote machine (e.g., on an NCAP).
- It is capable of handling local sensors directly connected to the Tablet PC and remote sensors connected to an NCAP or other networked computers. Sensors in this sense can be a sensor to be calibrated or a calibration standard.
- It is capable of connecting sensors using RS232 cables, an 802.11b wireless LAN, and Bluetooth.
- It can be easily operated by one person.
- It is capable of calibrating more than one sensor at a time. This feature can be used in a calibration facility where multiple pressure sensors are connected to a single pressure source manifold.
- It automatically computes calibration constants (slope and offset).

- It is able to retrieve the current calibration constants, and write the new constants directly to the smart digital sensors.
- It provides an effective Graphical User Interface (GUI) for displaying the sensor readings and accepting user commands.

B. FUTURE WORK

Since the calibration system was developed using a modular approach, it can be easily extended to accommodate other calibration requirements. The calibration system in the present form was mainly developed for the Honeywell PPT smart sensor. The procedures documented in this thesis can be used to develop calibration systems for other digital sensors.

The calibration system developed in this thesis is currently a stand-alone system. One possible direction for future work is to integrate it with the Integrated Condition Assessment System (ICAS).

The newly released LabVIEW 7.0 version provides support for handheld devices such as the iPAQ Pocket PC. It is possible to port the calibration system currently developed for the Tablet PC/Laptop PC to the Pocket PC.

LIST OF REFERENCES

1. R. Rupnow, J. Walden, X. Yun, D. Greaves, and H. Glick, "New calibration standings for next generation ship's monitoring systems," *Proceedings of the Thirteenth International Ship's Control Systems Symposium (SCSS)*, Orlando, Florida, April 2003.

2. R. J. Hogan, T. Cesarone and C. Savage, "Wireless expansion and enhancements of ICAS," *Proceedings of the Thirteenth International Ship Control Systems Symposium (SCSS)*, Orlando, Florida, April 2003.

3. Honeywell, ADS-14052 Rev. 11/00, "Precision Pressure Transducer PPT and PPTR," *User's Manual Version 2.4*, Honeywell, Inc., Plymouth, Minnesota, 2000.

4. OTC Wireless, "802.11b Wireless Serial Port Adapter," *User's Guide*, Revision 1.08, Fremont, California, May 2003.

5. S. J. Perchalski, "Shipboard sensor closed-loop calibration using wireless LANS and DataSocket transport protocols," Master's Thesis, Naval Postgraduate School, Monterey, California, June 2003.

6. 3e Technologies International, *3e-550I W-LION Industrial Wireless IO Node*, 3e Technologies International, Rockville, Maryland, 2002.

7. 3e Technologies International, *3e-521N Series Wireless Dual Mode Gateway User's Guide*, 3e Technologies International, Rockville, Maryland, 2002.

8. Crystal Engineering Corporation, *XP2 Digital Test Gauge Operational Manual*, PN: 2975 Rev A, CRYSTAL Engineering Corporation, San Luis Obispo, California, February 2003.

9. SI Pressure Instruments Ltd., *PC6 Operating Manual*, [www.si-pressure.com], Birmingham, B33, OYA, United Kingdom.

10. C. L. Lawson and R. J. Hanson, *Solving Least Square Problems*, Prentice-Hall, 1974.

11. C. Vern, "Sea Power 21, Projecting decisive joint capabilities," *Naval Institute Proceedings*, October 2002.

12. Noguchi, Yuki, "Rockville firm hoping to help the Navy go wireless," *Washington Post*, 07 May 2002.

13. 3e Technologies International, *3e-250 Bluetooth to RS-232 Cordless Adaptor*, 3e Technologies International, Rockville, Maryland, 2002.

14. D-LINK, Rev.2.0, *DBT-120 USB Bluetooth Adaptor*, D-LINK, Taiwan, August 2002.

15. Belkin, *Connecting people with technology*, [http://catalog.belkin.com], May 2003.

16. OMEGA Engineering Inc., *PX202, PX203, PX205, PX212, PX213, PX215 Pressure Transducers M2165/0395*, Omega, Stamford, Connecticut, 1995.

17. OMEGA Engineering Inc., *Your one-stop source for process measurement and control!,* [http://www.omega.com], May 2003.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Xiaoping Yun, Code EC/YX
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Fotis Papoulias, Code ME
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California

6. Jeff Walden
   NSWC Corona
   Corona, California

7. Randy Rupnow
   NSWC Corona
   Corona, California

8. LCDR Eusébio Pedro da Silva
   Ramos, Rio de Janeiro, RJ
   Brazil, CEP: 21060-130

9. Diretoria de Engenharia Naval
   Centro, Rio de Janeiro, RJ
   Brazil, CEP: 20010-000